CSE6730 - Final project report
World of Crypto: Is it still an uncharted territory?

Team 2
Aarohi Shah, Prathik Kaundinya, Saurabh Doodhwala

Github link: https://github.gatech.edu/pkaundinya3/Modeling-crypto-prices

April 2021

Abstract

Due to major financial crisis since the last three decades, there has been a strong motivation to study and research financial models to maximise the returns and minimise the risks for the future. A lot of work has been done in this domain and people have come up with numerous mathematical models to explain the future. Geometric Brownian Motion Model is one such model which has gained lots of attention and success for future stock predictions. Similarly, the model has also been successful to explain the prediction of Crypto-Currency prices for the future, which has gained a lots of attention in the last decade. In present work, we have tried to develop price prediction model of various cryptos like Bitcoin, Ethereum, Litecoin, Stellar, etc using Geometric Brownian Motion. Our goal is to use the historical data available for the cryptos along with an empirical model surrounding the news (attention) they gather to come up with an accurate prediction of the price. Our findings indicate that the GBM based model can predict the future cryptocurrency price with a reasonable accuracy. Finally, we have also designed a simulator to depict a particular long-term trading process that a customer might be interested.

1 Introduction

Cryptocurrency is a type of digital asset that is a digital currency (intangible) that uses a highly sophisticated type of encryption (cryptography) to secure and verify transactions as well as to control the creation of new units of currency. It is appealing to many users due to multiple reasons such as its highly secure underlying blockchain technology, anonymity, ease of transaction, and reduce long-term inflation associated with central banks. One of the well-known cryptocurrency is Bitcoin (BTC), which has gained a lot of attraction in the recent past. Other major types include Ethereum (ETH), LiteCoin (LTC), and Ripple(XRP).

As cryptocurrencies have exploded in value, there is an increasing interest and effort to understand them. Specifically, one of the interest areas is to model the behavior of the price of a particular cryptocurrency and predict future prices for potential capital investment. Considering that the world of crypto is relatively new, considerable amount of research has been done in network analysis [1, 2], however limited work has been done on predicting and comparison of future price for different currencies. Hence, in this project, we model and investigate price trends in cryptocurrencies using statistical techniques.

In general, predicting stock prices is difficult due to its highly volatile nature. However, it has been studied extensively in the literature using different modeling techniques such as mathematical models like Brownian motion [3, 4, 5], jump diffusion models [6], the Black-Scholes model [7] and AI driven models using unsupervised and supervised learning techniques [8]. Analogously, cryptocurrency prediction offers a lot of challenges due to its relatively fresh phase, unstable nature [9] and lack of fundamental intrinsic value components like P/E ratio, earnings performance, etc. (which are available for stocks). This allows the model to be purely defined by trade volume, and hence less predictable.

In this work, we propose to develop a price prediction model for cryptocurrencies using the geometric Brownian motion. We will use concepts like *random walk* and *Ito's process* for the construction and implementation of the geometric Brownian motion along with associated model statistics. Following this, we have presented an approach to include the attention (due to news or external factors) facor that a particular crypto is getting to account for its effect on the corresponding price. To demonstrate the aproach, we also evaluate the performance of our model by comparing he model predictions with actual values for three cryptos. Finally, we develop an index simulator to execute realistic trading scenarios considering the three selected cryptos.

2 Description

In this section we give an overview of cryptocurrency and some of mathematical processes that are suited to describe its behavior.

2.1 Definition

Cryptocurrency is often referred to as *decentralized money*. This implies that it is stored, created, and processed outside of a central bank, or government. More commonly known as coins, cryptocurrencies are digital currencies that are secured through cryptography. Most of them hence, rely on blockchain technology which is a distributed ledger of all transactions that is decentralized and unable to be changed under most circumstances as long as nobody controls more than 50% of the computing power on the network.

2.2 Efficient market hypothesis

The dynamics of a stock price shows uncertain (random) movements of their value over time. This is mainly attributed to efficient market hypothesis (EMH). Recent studies have shown that Bitcoin also follows the EMH [10]. The EMH states that: (1) The present value of an asset reflects its past history, (2) Markets respond almost immediately to any new information / news about that asset. From these two properties, it may be inferred that the the price of a coin follows a Markov process. This makes the representation of the predictions in terms of probability distributions and helps to model the currency with a certain level of confidence.

2.3 Cryptocurrency process

For the purpose of this project, we are going to model the behaviour of Cryptocurrencies to understand appropriate buy/sell times to maximize returns. Hence, it is first important to understand the behaviour of the cryptocurrency as a process. In this section we provide description of all the necessary mathematical concepts, required to develop and analyse a cryptocurrency model. There are two steps to model the future price of a financial asset: i) Modeling the new information ii) Assuming a probability distribution on the observed data. From literature [4], we know that the price behaviour of asset shows similarity with behaviour of stochastic process called *Brownian Motion*.

Brownian motion Brownian motion is typically used to model the motion of any time series variable and dates back to the nineteenth century where it was used to understand the movement of particles in water [11]. Since then such a model has been used to simulate and investigate advanced phenomena a diverse range of fields. Researchers have also explored Brownian motion for stocks markets/financial asset modeling and prediction due to the random changes in these asset prices. In the context of financial markets, assets have a continuous price which evolves with time and hence are driven by Brownian motion processes [12, 13]. Such a model uses the concept of *random walks* to exemplify its stochastic nature. Here, we model the price prediction of cryptocurrency using Brownian Motion. The two properties of Brownian motion that will aid in developing the modeling are:

- **Continuity:** If process B is following Brownian Motion, then B has a continuous path over each point of time. This is important because the price of cryptocurrencies exist at each point of time.
- **Markov Property:** The probability distribution of the process B at time $'t'$ depends only on the probability distribution of the process B at time $'s'$, where $s < t$.

Generalized Random Walk The generalized random walk is also known as Brownian motion with a drift component and is used for this work. Considering a stochastic process B_t and certain constants μ and σ , we can describe this process as [12]:

$$B_t = \mu t + \sigma W_t \tag{1}$$

such that $W_t = \epsilon\sqrt{t}$, (ϵ : random number from standard normal distribution) is a random walk process [14]. We will describe this process in the context of this work in a detailed manner for the final report. This process has a mean of μt and variance of $\sigma^2 t$, which represent the *drift* and the *volatility* of the process.

3 Base Model

Investors are typically interested in predicting a profit or loss realized by trading particular financial asset. Cryptocurrencies are also viewed with a similar lens. Hence, it is practical to make conclusions about the relative changes in time (increase or decrease) of a crypto. We use this fact to model the return of a cryptocurrency and hence its behavior over time using a stochastic differential equation analogous to that used for stocks [4].

3.1 Conceptual Model

Mathematical model Let us consider, S_t to be the value of the cryptocurrency at time t with an expected rate of return μ, then the change in value over the next considered time period has 2 components: (1) deterministic/expected component which is the expected return from the crypto during a period dt: $\mu S dt$, (2) stochastic/unexpected which is rather tough to deal with and deals with the erratic changes during dt that occurs due to external events. We can hence, address this stochastic component of the price using the modeling and simulation approach and with a Brownian process. As a first attempt, we assume this to be $\sigma S_t dB_t$, that is this contribution is also proportional to the value of that particular crypto. Overall, this leads to a mathematical model in the form of a stochastic differential equation given by:

$$dS_t = \mu S_t dt + \sigma S_t dB_t \tag{2}$$

Solving the mathematical model The solution to the described mathematical model can be computed using the Ito's [15] formula. The main concept in Ito's calculus is Ito's stochastic integral, which is given as:

$$Y_t = \int_0^t H dB$$

Here, B is a Brownian Motion and H is square-integrable process generated by X.The integrand represents the amount of the asset held, the integrator represents the movement of the prices of the asset, and the integral is the total capital available (including monetary value of the asset) at any given moment. Ito's stochastic integral represents the payoff of a continuous-time trading strategy consisting of holding an amount H_t of the asset at any time t. In this situation, the condition that H is adapted corresponds to the necessary restriction that the trading strategy can only make use of the available information at any time. This prevents the possibility of unlimited gains through high-frequency trading: buying the stock just before each up-tick in the market and selling before each down-tick.

In the context of this work, we can use the general form of Ito's formula given by:

$$dH = \left(\frac{\partial H}{\partial X} a + \frac{\partial H}{\partial t} + \frac{1}{2}\frac{\partial^2 H}{\partial X^2} b^2\right) dt + \frac{\partial H}{\partial X} dB \tag{3}$$

such that $dX = adt + bdB_t; a, b :$ constants. Hence, if we consider $H = ln(S_t)$, we get:

$$\frac{\partial H}{\partial S} = \frac{1}{S_t}, \frac{\partial^2 H}{\partial S^2} = -\frac{1}{S_t^2} \text{ and } \frac{\partial H}{\partial t} = 0 \tag{4}$$

Using, Eqn. 4 in 3, we obtain:

$$d(lnS_t) = ln\left(\frac{S_t}{S_{t-1}}\right) = \left(\mu - \frac{1}{2}\sigma^2\right) dt + \sigma \epsilon \sqrt{dt} \tag{5}$$

It follows that $d(lnS_t)$ has a brownian motion with drift and hence, thee solution can be drawn as:

$$S_t = S_{t-1} e^{(\mu - 0.5\sigma^2)dt + \sigma\epsilon\sqrt{dt}} \tag{6}$$

Eqn. 6 represents the solution to model a cryptocurrency value at time S_t using the initial value S_0 of the period dt. It is worth mentioning, that for parameter estimation we use the training set to compute μ and σ.

Cryptocurrency distribution In this work, we use the log normal probability distribution to model the price of an asset at a future time. A log normal distribution over an RV (random variable) R if another RV $Q = log(R)$ follows a normal distribution. Let Q be parametrized as $Q \sim \mathcal{N}(\mu, \sigma)$. Then, the probability density function (PDF) of the lognormal distribution of R is expressed as:

$$f(R) = \frac{1}{\sqrt{2\pi\sigma}} exp\left(-\frac{1}{2}\left(\frac{R-\mu}{\sigma}\right)^2\right)$$

There are two primary advantages to interpreting the future value of an index as a probability distribution: (1) A robust uncertainty quantification for a prediction (i.e., predictive variance is well defined) and (2) The expected value is straightforward to compute.

Expected Cryptocurrency Value Based on the definition of the considered probability distribution and considering the defined mathematical model, we obtain $R = log(S_t)$ such that R is normally distributed with a mean of $log(S_0) + (\mu - 0.5\sigma^2)t$ and a variance of $\sigma^2 t$. Hence, the expected value $E(S_t)$ at a future time is given by:

$$E(S_t) = e^{log(S_0)+(\mu-0.5\sigma^2)dt+\sigma^2 t} = S_0 e^{(\mu-0.5\sigma^2)dt} \tag{7}$$

3.2 Validation

3.2.1 Implementation details

We perform various simulations to analyse the model accuracy for test data. Parameters, namely drift and volatility are estimated for the train data which is then used to simulate the future values of a cryptocurrency for a certain period of time. We first show results and analysis for BitCoin (BTC) since maximum historical data is available for this crypto. Later we also show, major results for other currencies. Further, we use two methods to implement the conceptual model:

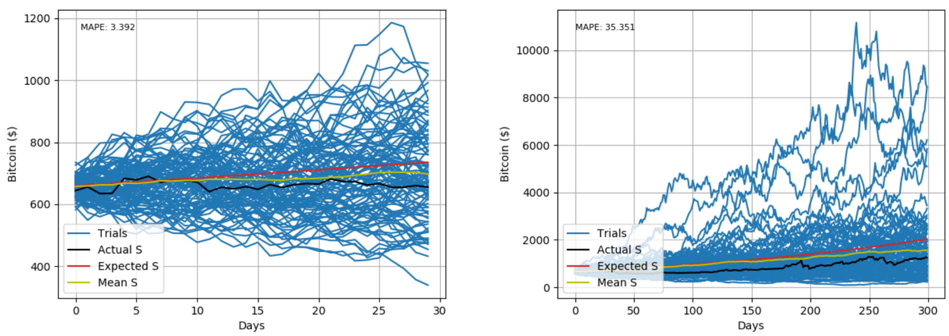
1. Single model: Uses training set of define period to estimate the model parameters. This parameters are used for the entire prediction case. For instance, if our sample sequence is 300 days, we generate 300 random numbers (ϵ) and use one set of parameter models to estimate $S_t + 1$

2. Rolling model: In this case we use a smaller time period (say $N = 30$ days) to estimate parameters and simulate S. Then, S_N is used as the S_0 for the next considered period, thereby having a *rolling* nature.

We show results for both these approaches using a maximum of 300 days to present the advantages and disadvantages/limitations associated with both. Also, Maximum Absolute Percentage Error (MAPE) is used as the quantitative error metric to represent accuracy of the model.

3.2.2 Single model

For the single case, we use 30 days to 300 days to analyze accuracy. Figure 1 shows plots for $N = 30$ and $N = 300$ whereas Fig. 2 shows the MAPE obtained for all simulated cases. We observe that as the considered time period (N) increases, the MAPE also increases, implying that the model accuracy is reducing. This is intuitive, since as N increases, the data that the model parameters has to represent, significantly increases. This becomes difficult for two parameters to be represented thereby, averaging the possible scenarios, and sacrificing accuracy. That said, we believe an MAPE of $< 20\%$ is reasonably good, considering the simplicity of the model. It is worth mentioning that the number of prediction paths considered for all the cases until now is 100. We also investigated the results by increasing the number of paths, however it does not have much effect in this case.



(a) N=30 days

(b) N=300 days

Figure 1: Sample simulations results for single case as compared to actual data

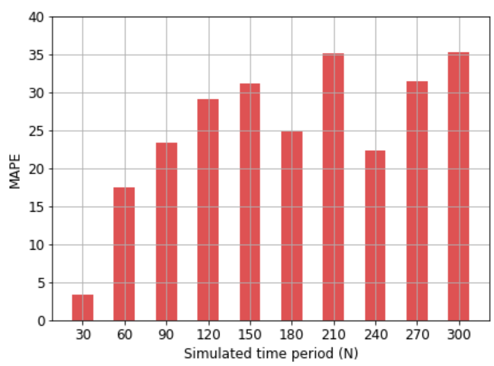


Figure 2: Representative model accuracy: MAPE for different N

3.2.3 Rolling model

As shown in the single model, the resulting error is high when the model is simulated for larger period of time. Moreover, what is observed that for N less that 60 days, we obtain reasonable accuracy. Hence, here we show results for a rolling model, which uses results of the previous selected period of same size to predict for the current time window and so on. For the results, we consider a total simulation time of 300 days which was also used in the single model, for systematic comparison. So, if a rolling period (R) of 30 days is used, there are a total of 10 successive such periods simulated.

In Fig. 3, we show the case of simulating 300 days with 3 periods, yielding a value for R as 100 days. An average MAPE of 16.5 is obtained which shows a drastic improvement from the single case simulation of 300 days. We can leverage these findings, to further explore different rolling periods and analyze the accuracy associated with them. Figure. 4 shows the average MAPE obtained for 300 days by varying R from 30 to 100 (plots for simulations for all cases are added to Appendix). We obtain reasonably good results when R is 30 days, which seems practical in terms of implementation also. In other words, it does seem practical to simulate for 30 days at a time and using the predictions for the next 30 days. Note that this method, however heavily relies on the prediction of the previous period and hence there are high chances of error propagating through time. Hence, it is important to appropriately pick the rolling period to avoid such a scenario.

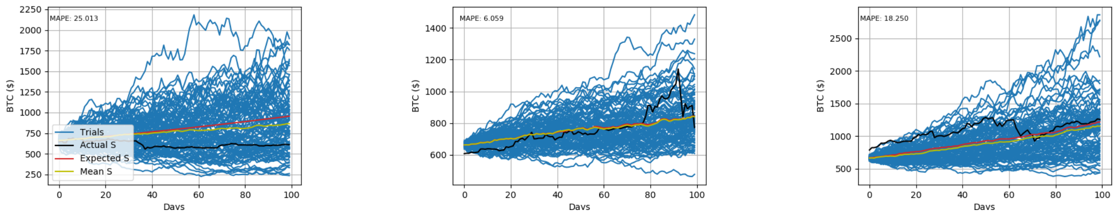


Figure 3: Sample rolling model simulation result for rolling period: 100 days

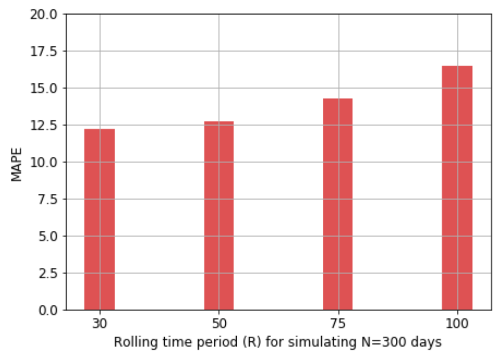
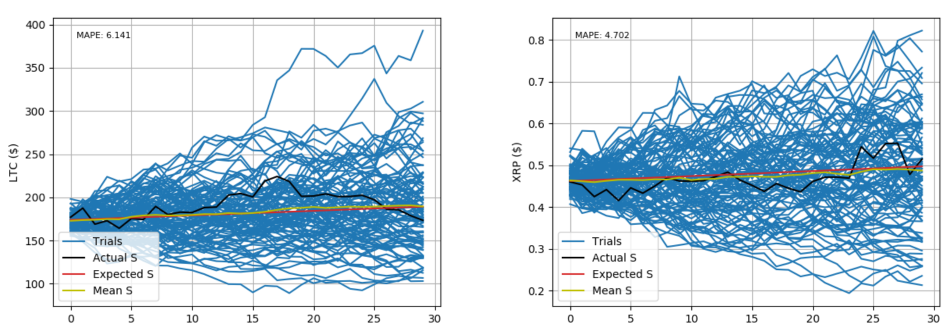


Figure 4: Representative model accuracy: MAPE for different R in the rolling case for N=300 days

3.3 Different cryptocurrencies

With the objective of having a simulator that considers a portfolio of 3 cryptcurrencies, we also evaluate the accuracy of the developed model for two other crytos namely, LTC and XRP and show a sample prediction in Fig. 5.



(a) LiteCoin prediction

(b) Ripple prediction

Figure 5: Sample simulations results for other cryptocurrencies as compared to actual data for N=30 single case

As we can see, the accuracy of the model is very good with an MAPE< 10% for both cases, inspite of

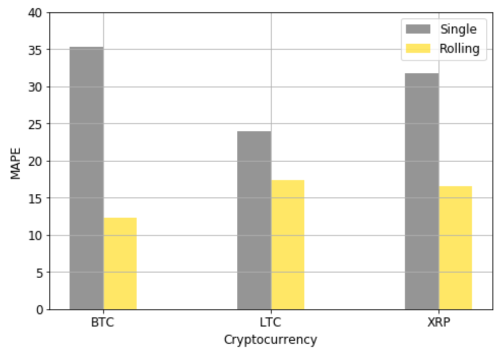


Figure 6: Comparison for single and rolling for 3 cryptocurrencies

less data being available. We also show a comparison of the rolling and single approaches for all the three considered cryptos with N=300 days for both cases and R=30 days for the rolling cases in Fig. 6. Overall, we observe that rolling approach improves the prediction accuracy. The improvement is different for different cases due to the difference in the amount of data available as well as the inherent behavior of that data (steep vs. smooth changes over time).

4 Attention model

Based on the results and conclusions obtained above, we explore an additional parameter that may have a significant impact on the accuracy of the model predicting the future price of a particular cryptocurrency. We utilize a parameter, called the attention parameter, which is known to drive the BitCoin price [16, 17, 18, 19]. There are many possible ways of describing this attention parameter, such as number of Google searches, Wikipedia requests, or more traditional factors such as volume of transactions. For this work, we represent the attention parameter as a function of the number of Google searches for a cryptocurrency (available at trends.google.com). The description of this model along with the updated conceptual model definition is given in the following paragraph.

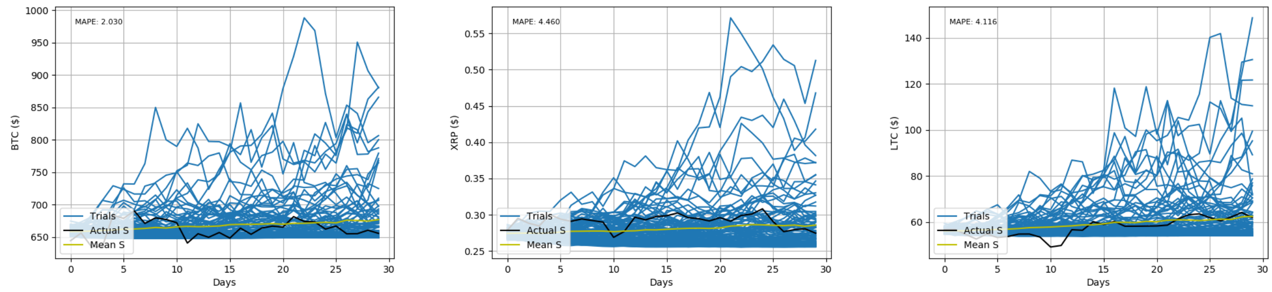
Updated mathematical formulation: Keeping all the variables from the basic geometric stochastic model consistent, we consider P to represent the attention parameter which also follows a Brownian process. Hence, the overall model can be represented by:

$$dS_t = \mu P_{t-\tau} S_t dt + \sigma \sqrt{P_{t-\tau}} S_t dB_t \tag{8}$$

$$dP_t = \mu_p P_t dt + \sigma_p P_t dZ_t \tag{9}$$

such that μ_p and σ_p are the drift and volatility for the attention parameter respectively. Z_t is the brownian process behind the attention parameter behavior due t ts random nature. It is important to note that, once factor P is estimated using Eq. 9, it is used to scale the drift and volatility coefficients in Eq. 8. Also, we consider a possible delay or lag (τ) that may exist between the increase or decrease in the attention parameter and its effect on the price of the considered cryptocurrency. For all the results presented in this work a lag of 2 days was considered, to include more dynamic effects on the price, versus having a dampening effect. Note that, we perform all the parameter estimation and model solution methods based on the methodology used for the basic model described in Sec. 3.

Cumulative Results Here, we first compare the price prediction of the cryptocurrency by including the attention parameter. Further, we use the single case implementation technique to simulate for 30 days. The results for all three cryptos have low errors (less than 5%) as shown in Fig 7.



(a) Bitcoin prediction (b) Ripple prediction (c) Litecoin prediction

Figure 7: Sample simulations results for other cryptocurrencies as compared to actual data for N=30 single case using attention parameter

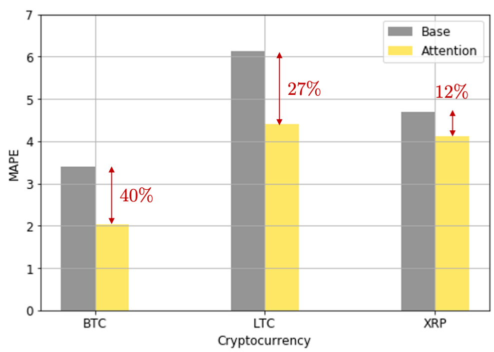


Figure 8: Base model vs. attention model for different cryptos

It is imperative we analyze the results in terms of the base model in order to understand the relative effect of the attention parameter of prediction accuracy. We do this in Fig 8, wherein we compare the MAPE obtained for single case of 30 day prediction with and without including the attention factor. We can see that a very good improvement (40%) is obtained for BTC when using attention versus a 27% for LTC and 12% for XRP. This leads to multiple conclusions: (1): Each crypto has a nonlinear behavior in terms of the attention parameter, implying that considering the popularity of BTC, the attention parameter has more effect on its value versus say XRP which is not that popular. (2): Based on the above conclusion, it is important to realise that, even though not considered here, there might be other currencies that are not that popular, whose prediction using the presented model may lead to a compromised accuracy. Hence, for such cases, it might be useful to use only the base model which directly models the price based on previous trends and an appropriate probability distribution.

Finally, we also conducted the simulation by using the attention parameter for longer terms of invesstment such as 1 year. However, we observed that the accuracy of for the model drastically reduces. This in a way is intutive as the attention paramter uses a lag of 2 days and it is difficult for the predicted parameters related to attention, to represent a very long period of time. In other words, if the news shows that Tesla invested in BTC, its effect on BTC price can be accounted for a few weeks by will eventually die of over the year. A rolling implementation scheme as discussed earlier might come to rescue, but not in its current form. Since, in the current form, we only consider google searches to represent the attention. But in order for the rolling scheme to work accurately, more information pertaining to attention need to be considered since, this trend is not totally random. More discussion along with future directions are given in Sec. 6.

5 Simulator for cryptocurrency portfolio

5.1 Framework

We develop a cryptocurrency index simulator to evaluate the performance of our model on a fictitious portfolio consisting of $100,000. Overall framework for the simulator design is shown in Fig. 9.

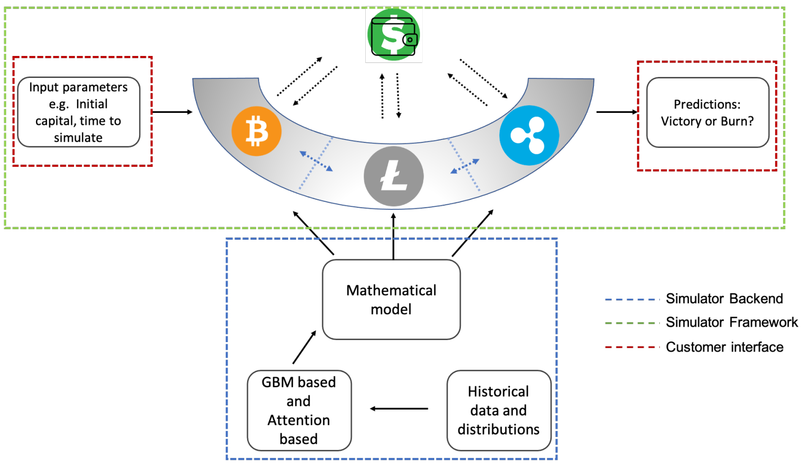


Figure 9: Simulator design for crypto portfolio

More specifically, we impose the following constraints on our model during the simulation:

- The portfolio will be restricted to making trades on the 3 cryptocurrencies mentioned.
- The portfolio can never open a short position (i.e., sell currency not owned).
- At every instant of time t, only a single trade action is taken - either a buy of one of the currencies, the liquidation of the current holdings in one of the indices, or a hold (no action taken).
- The chosen action is dependent on the values of the expected percentage price difference δ. For each of the currencies, we have the equations:

$$\delta_B = \frac{B_{t+1} - B_t}{B_t} * 100$$

$$\delta_X = \frac{X_{t+1} - X_t}{X_t} * 100$$

$$\delta_L = \frac{L_{t+1} - L_t}{L_t} * 100$$

  where, B, X and L represent prices of the three currencies. The *action* is determined using this framework:

  - The largest expected loss n_{max} is defined as

$$n_{max} = \begin{cases} min\{\delta_B, \delta_X, \delta_L\}, & \text{if } min\{\delta_B, \delta_X, \delta_L\} < 0 \\ 0, & \text{otherwise} \end{cases}$$

– The largest expected gain p_{max} is defined as

$$p_{max} = \begin{cases} max\{\delta_B, \delta_X, \delta_L\}, & \text{if } max\{\delta_B, \delta_X, \delta_L\} > 0 \\ 0, & \text{otherwise} \end{cases}$$

– We define a predetermined *hold threshold* h, which allows the chosen action to be computed as

$$\text{Action } = \begin{cases} \text{buy}, & \text{if } |p_{max}| > |n_{max}| \text{and } p_{max} > h \\ \text{sell}, & \text{if } |p_{max}| < |n_{max}| \text{and } |n_{max}| > h \\ \text{hold}, & \text{otherwise} \end{cases}$$

Note that the buy and sell operations are performed on the holdings corresponding to $max\{\delta_B, \delta_X, \delta_L\}$ or $min\{\delta_B, \delta_X, \delta_L\}$, respectively.

– If a buy is not possible (i.e., the portfolio balance is completely allocated) the simulation holds if a buy decision is made. Likewise, if there is no current holding of the currency with a sell order, the portfolio is held at the current state.

- The simulation ends with either a *victory*: defined as being able to pass a certain number of time steps without exhausting capital or *defeat*: burning through initial capital.

5.2 Tutorial: Sample Run

A sample simulator run for the base model with the rolling implementation scheme is described here. This can be easily applied to the attention model also with certain modeling modifications as discussed in Sec. 6. In order to run a simulation, *simulator.py* needs to be run with the input parameters shown in Table. 1. The corresponding values in this table are only sample ones to show here. Customers are encouraged to try different values depending on the allowable range (embedded in simulator.py).

Parameter	Variable	Value
Currencies to be considered (max=3)	currencies	btc,xrp,ltc
Simulation Start date	start_data	2018-05-01
Number of trades	time_steps	20
Trade interval	timedelta	15 (days)
Minimum percentage threshold for action	h	0.1
Initial capital amount	start_capital	100,000 USD
Allowable Trade amount/trade	trade_amount	5000 USD

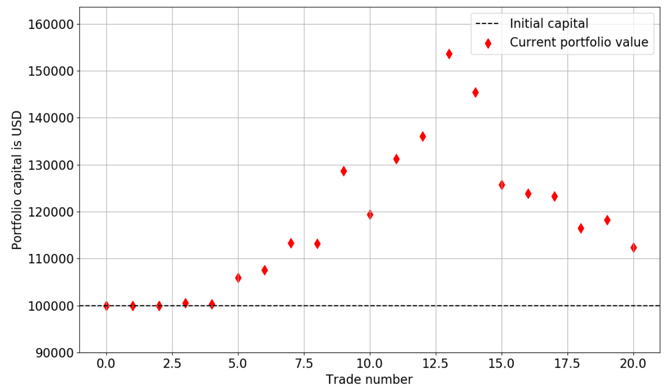
Table 1: Simulation parameters for tutorial

Note that, in Table. 1, a value of 20 is used for time_steps, which implies that 20 trades will be simulated at intervals of 15 days (timedelta). This enables a simulation for prediction over a total of $20 \times 15 = 300$ days. Hence, these parameters can be adjusted based on custtomer requirement, that is if the customer is interested in short/long term holdings and/or frequency of tradings. Also, threshold parameter (h) is important, since a too high value can result in conservative simulations (as the simulator will not pick/ drop a particular crypt unless the change in its value is correspondingly high).

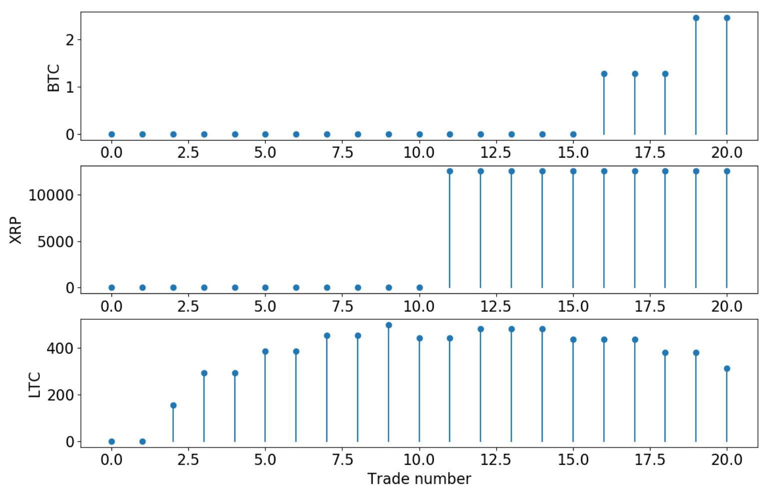
While running the simulator runs with the user-defined input parameters, it is designed such that it outputs the summary of the trade and portfolio after every time interval. Hence, if timedelta is 15 days, essentially the simulator reports the trades and holding after every 15 days from start date. At the end of all time periods, a summarized report is printed indicating the number of holding for each crypto and corresponding end value as shown in Fig. 10a. We also display a plot such as in Fig. 10b is displayed for the customer to look at the trend of the simulator predictions and its effect on the holdings. And finally,

```
Portfolio consists of:
btc balance = 2.454067769247283.........$10540.179940156007
xrp balance = 12560.376166644648.........$3069.8759187686205
ltc balance = 313.7420326570315.........$23786.821024383844
Total value of the portfolio is $112396.87688330848
```

(a) Sample portfolio report after simulation



(b) Trend of portfolio capital over simulated time



(c) Individual crypto holdings over simulated time

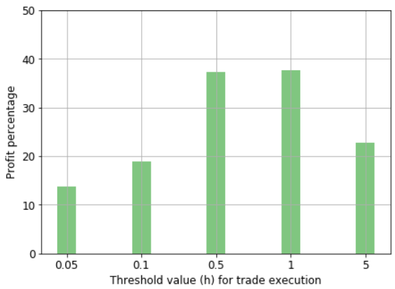
Figure 10: Sample simulation results at customer end

customers can also see the amount of indvidual crypto holdings in simulated for their portfolio as seen in Fig. 10c. This helps the customer get an idea/estimate of the number of cryptos or the distribution of different cryptos in their portfolio.

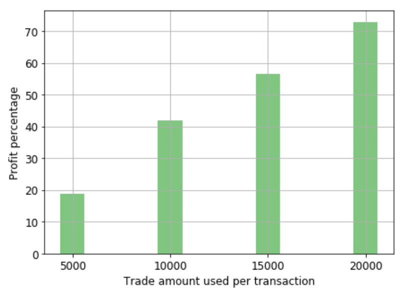
For this particular case, shown in Fig. 10, we get a net profit of approximately 12000 USD when simulated for a period of 300 days. Further, we observe that the simulator is able to predict with almost negligible loss during the overall run. Also, we see that BTC is not traded initially and that is probably attributed to two reason: (1) due to the low changes occurring in its value such that it doesn't pass the threshold for action (2): the model thinks that investing at that time in BTC may result in significant losses.

5.3 Sensitivity studies

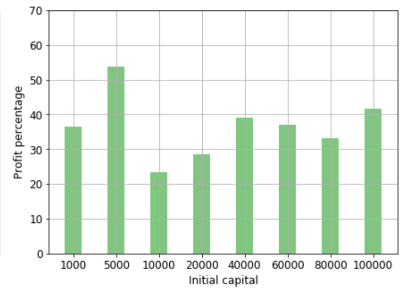
Using the designed simulator, we conduct sensitivity studies in order to understand the effect of various model parameters on the simulation predictions. The results for these studies are shown in Fig. 11. Note that in addition to allowing the simulator to cater to different types of customers, these studies also serve as a validation and verification for the simulator. For instance, if the simulator is predicting profits, these trends gives us an idea of whether the amount of these profits make logical sense as discussed in the subsequent paragraphs. The constant parameters considered for these study have values that are mentioned in Table 1, except that a 10 day of time interval is used with 30 time steps. These parameters remain constant depending on which parameters sensitivity is studied. Hence, when threshold value (h) is studied in Fig. 11a, all other parameters take the above mentioned values except h.



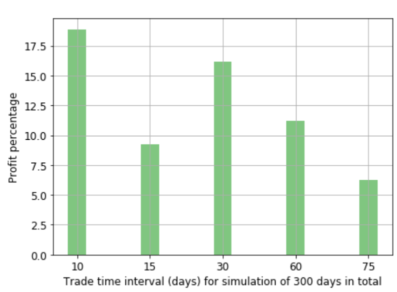
(a) Threshold value



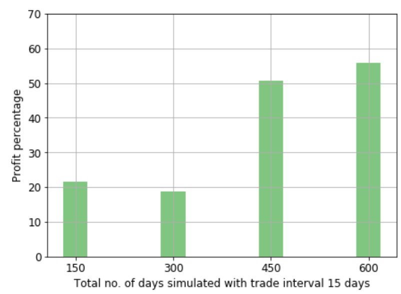
(b) Trade amount



(c) Initial capital



(d) Trade interval



(e) Total time for simulation

Figure 11: Simulator sensitivity studies

As explained earlier, the threshold parameter (h) that decides whether an action or trade occurs or not, is crucial. This also reflects in a way on the amount of risk that a customer is willing to take. For e.g. if a low value is used, a trade is made possible even if small changes in price occur over time. Hence, if we see in Fig. 11a, we obtain relatively lower profit when very low h values are taken. This is probably because extremely low crypto value changes also allow for trades which may result in some losses over time. On the other hand, if a very large value of h is chosen, a conservative approach is taken by the simulator, thereby again resulting in not best possible gains.

We conduct sensitivity studies for trade amount as well as initial capital in Figs. 11b and 11c. This was done, so that we can analyze the simulator for different types of customers. For instance, all customers may not have a lot of capital to invest, and hence we show that in such cases also that simulator can predict well with a predicted profit of 25-50%. Note that, when we study the sensitivity for initial available capital, we also scale the trade amount to a 10% value as a reasonable value for allowable trade. We also observed that, as we increase the amount of trade value allowed per transaction for an initial capital of \$100k, the profits seem to increase almost linearly in the beginning and stabilizing as we keep on increasing.

Finally, we show the simulator prediction sensitivity with respect to the time, i.e. in terms of trade time interval and total number of days for investment in Figs. 11d and 11e respectively. Through this we can see the predictions for customers who are interested in short-term and more dynamic trading versus customers who are interested in long-term investments and don't want to trade that frequently. We observe that the simulator performs the best, when more number of trades are allowed in a given time. This is intuitive since the simulator can leverage more recent information and tailor the portfolio holdings accordingly. This serves also as a verification for the simulator. Further, the simulator gives more profit when the capital is invested for a long period of time, which is also something that is typical of cryptocurrencies, given the current trends and profit averaging that occurs over long periods.

Overall, these findings can help different types of customers make decisions about their investments and observe the return they may get over different times.

6 Conclusions and Outlook

In summary, we modeled a price prediction framework for cryptocurrencies using a mathematical model based on the geometric brownian motion. We show two implementation schemes, namely single and rolling which enables a prediction for short and long periods of time with reasonable accuracy. The model is validated and verified by comparison of accuracy obtained for test data. We observed that the rolling scheme helps maintain low errors when predictions are made for long periods of time.

Following this, we have presented an approach to include the attention (due to news or external factors) factor that a particular crypto is getting to account for its effect on the corresponding price. We also present the update mathematical conceptual model fr this approach and evaluate its performance using the single scheme for implementation. We conclude that the effect of attention parameter is different of different cryptos, thereby necessitating a deeper look at the parameter to be considered for a particular crypto. The lag period considered is also an important factor and a correlation between the lag period and its effect of price can be carried out to understand the model dynamics in a better way. Most importantly, we observe that the above challenge of catering to different crypts and for a rolling scheme implemented for the attention based model, the attention parameter considered should be more than just the number of google searches. A parameter which includes more information pertaining to the value of the crypto itself needs to be included to make the attention index more substantial. One possible approach can be a parameter that is representative of the ratio of google searches or wikipedia requests to the actual number of trades (buys and sells) that occurred for that crypt in the considered time. Such an index may be able to represent the effect of attention: positive and negative more accurately.

Finally, we developed an index simulator to execute realistic trading scenarios considering the three selected cryptos. We also provided a sample tutorial to run a simulator which then allows the customer to visualize the profit made along with individual holdings and their respective values in the portfolio. We also conducted multiple sensitivity studies to analyze the simulator prediction with parameters such as initial capital, trade interval, investment period, etc. Based on our findings, we can conclude that, the simulator performs well in all cases and can cater to different types of customers. In other words, a group of students who are relatively new to the world of cryptos and want to invest 1000 USD can get predictions from our simulator and can help them make informed decisions as well as a trader who does short-term trading and has access to reasonably high amount of capital and is willing to take more risks in terms of trading amount can also reap benefits. Moreover, these sensitivity studies also serve as a validation and verification for the simulator since the trends obtained give us an idea of whether the amount of the predicted profits make logical sense. It is worth mentioning that, there are multiple future directions to scale the complexity and representation of the simulator. For instance, in its current form the simulator is designed to use a constant amount of trading value at each transaction or the threshold value for deciding whether the trade action occurs or not is constant. However, these parameters can be made dynamic depending on customer preference (dynamic changes on customer behavior for taking risks) or long term trend of crypto respectively.

To conclude, the field of cryptocurrencies and related research is gaining traction considering the current industry and political interest. To this effect, we presented a mathematical model and an index simulator which show reasonable accuracy, such that the simulator is able to predict significant profit returns for different types of customers and investments. This leads to a wide area of research which can make us believe that the world of cryptos will soon no longer be an uncharted territory.

7 Logistics

The code for the project was written using Python 3.7, with the set of dependencies specified in the requirements.txt file on GitHub.

We have divided the work in a fair way to include each members contribution. Due to the remote fashion of the class we have heavily relied on Bluejeans to discuss not just the conceptual model, but also implementation and coding of the developed model. From a development perspective, Aarohi and Saurabh have worked on the conceptual model, execution of basic model for various studies and Prathik has worked on the code. All members have met frequently for various discussions, whether about conceptual model or implementation of code. In a similar fashion, Aarohi and Saurabh have worked on the attention parameter modeling and implementation. Prathik and Aarohi have worked on thee simulator design and implementation. All members have worked fairly equally on the ideation, conceptualization, implementation and the associated deliverables (cleaining up the git and all reports).

GitHub details: The GitHub repo containing the source code for the project has been shared with the instructor and TAs for the class.
Link: https://github.gatech.edu/pkaundinya3/Modeling-crypto-prices

References

[1] Matthias Lischke and Benjamin Fabian. "Analyzing the bitcoin network: The first four years". In: *Future Internet* 8.1 (2016), p. 7.

[2] Annika Baumann, Benjamin Fabian, and Matthias Lischke. "Exploring the Bitcoin Network." In: *WEBIST (1)*. 2014, pp. 369–374.

[3] I Maruddani Di Asih et al. "Modeling stock prices in a portfolio using multidimensional geometric brownian motion". In: *Journal of Physics: Conference Series*. Vol. 1025. 1. IOP Publishing. 2018, p. 012122.

[4] Abdelmoula Dmouj. "Stock price modelling: Theory and Practice". In: *Masters Degree Thesis, Vrije Universiteit* (2006).

[5] Sorin R Straja. "Stochastic modeling of stock prices". In: *Montgomery Investment Technology, Inc* (1997).

[6] Steve G Kou. "Jump-diffusion models for asset pricing in financial engineering". In: *Handbooks in operations research and management science* 15 (2007), pp. 73–116.

[7] AS Shinde and KC Takale. "Study of Black-Scholes model and its applications". In: *Procedia Engineering* 38 (2012), pp. 270–279.

[8] Sidra Mehtab, Jaydip Sen, and Abhishek Dutta. "Stock price prediction using machine learning and LSTM-based deep learning models". In: *arXiv preprint arXiv:2009.10819* (2020).

[9] Marie Briere, Kim Oosterlinck, and Ariane Szafarz. "Virtual currency, tangible return: Portfolio diversification with bitcoin". In: *Journal of Asset Management* 16.6 (2015), pp. 365–373.

[10] Bartos Jakub. "Does Bitcoin follow the hypothesis of efficient market". In: *International Journal of Economic Sciences* 4.2 (2015), pp. 10–23.

[11] Robert Brown. “XXVII. A brief account of microscopical observations made in the months of June, July and August 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies”. In: *The philosophical magazine* 4.21 (1828), pp. 161–173.

[12] Tsekov Roumen. “Brownian markets”. In: *Chinese Physics Letters* 30.8 (2013), pp. 88901–088901.

[13] Angeliki Ermogenous. “Brownian motion and its applications in the stock market”. In: (2006).

[14] Barry D Hughes. *Random walks and random environments. 2. Random environments.* Clarendon Press, 1996.

[15] Marc Revuz Daniel; Yor. *Continuous martingales and Brownian motion.* Springer, 1999.

[16] Jaroslav Bukovina, Matus Marticek, et al. “Sentiment and bitcoin volatility”. In: *MENDELU Working Papers in Business and Economics* 58 (2016).

[17] Pavel Ciaian, Miroslava Rajcaniova, and d’Artis Kancs. “The economics of BitCoin price formation”. In: *Applied Economics* 48.19 (2016), pp. 1799–1815.

[18] Ladislav Kristoufek. “BitCoin meets Google Trends and Wikipedia: Quantifying the relationship between phenomena of the Internet era”. In: *Scientific reports* 3.1 (2013), pp. 1–7.

[19] Gianna Figa-Talamanca and Marco Patacca. “Does market attention affect Bitcoin returns and volatility?” In: *Decisions in Economics and Finance* 42.1 (2019), pp. 135–155.

Appendix

Supplementary results for rolling case

We show the prediction results for rolling period of 30, 50 and 75 days such that total considered time is constant (300 days) in Figs. 12, 13 and 14.

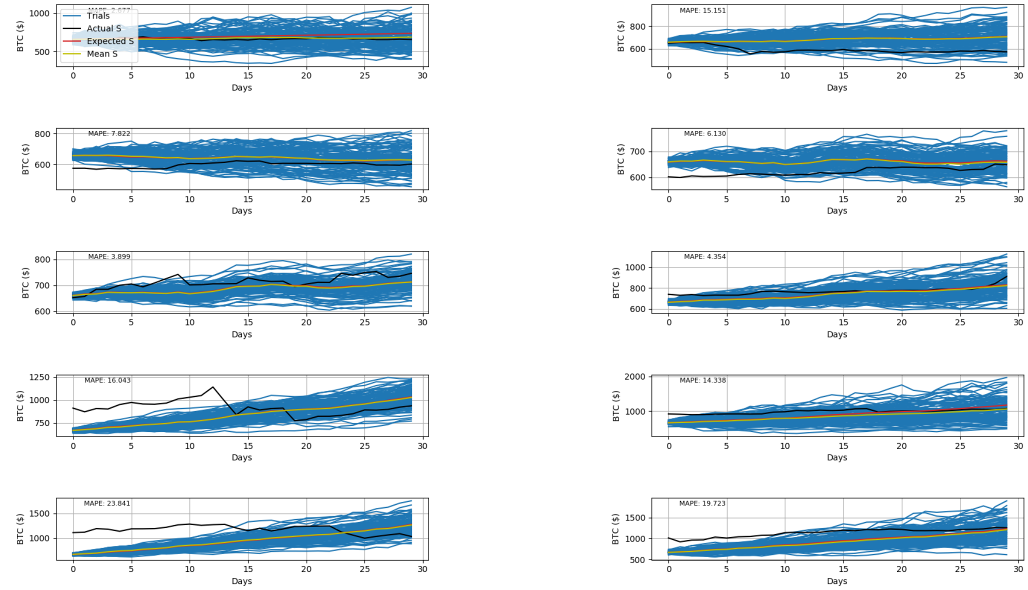


Figure 12: Rolling model simulation result for rolling period: 30 days

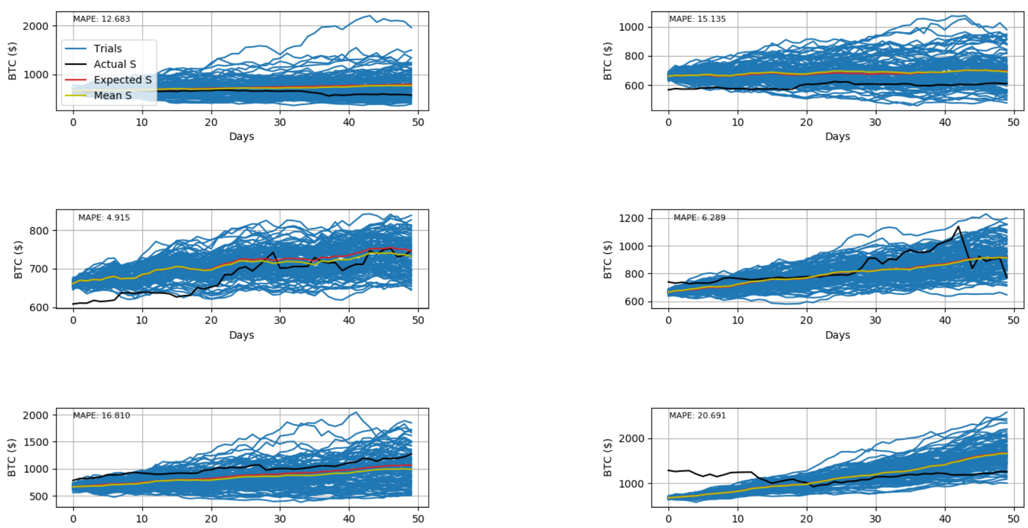


Figure 13: Rolling model simulation result for rolling period: 50 days

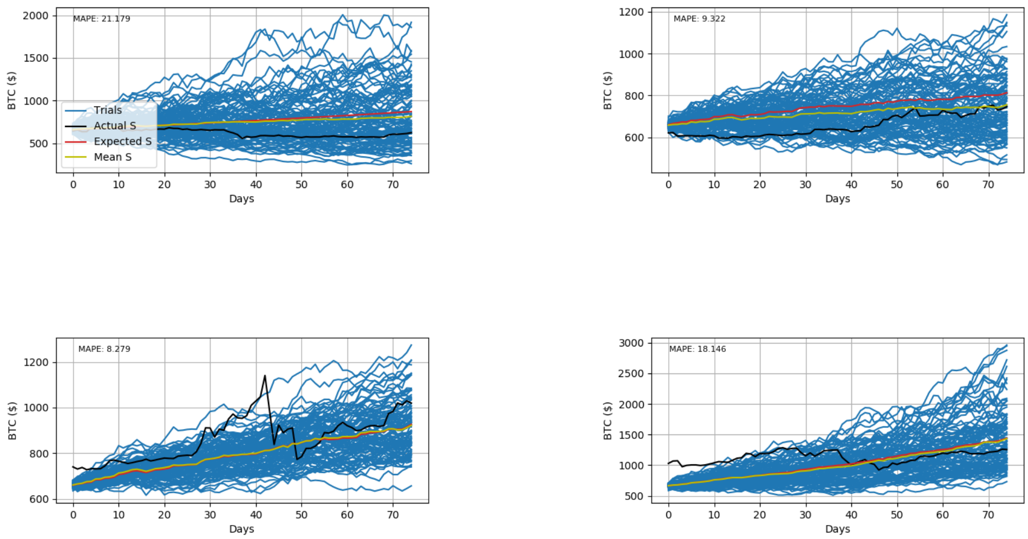


Figure 14: Rolling model simulation result for rolling period: 75 days

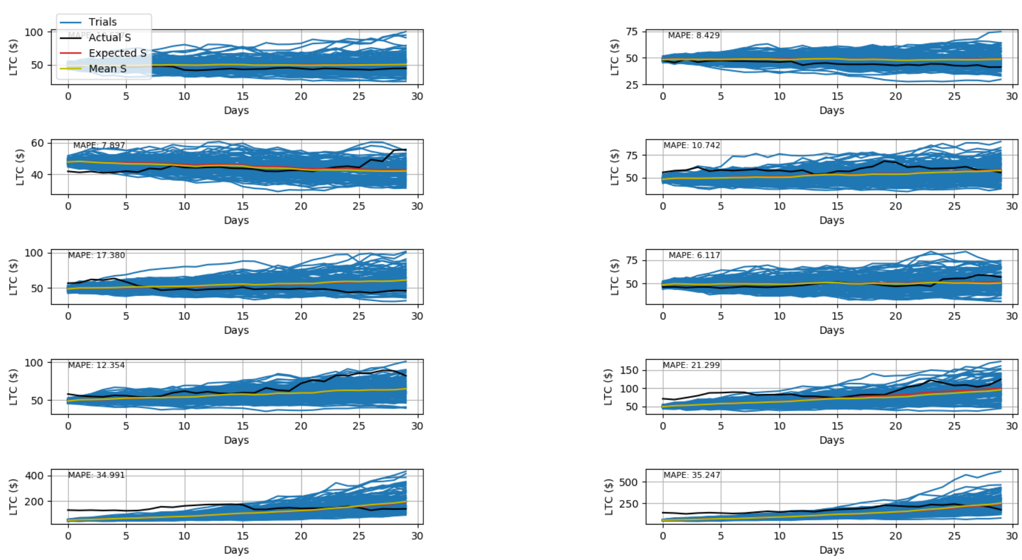


Figure 15: Rolling model simulation result for LiteCoin with rolling period: 30 days

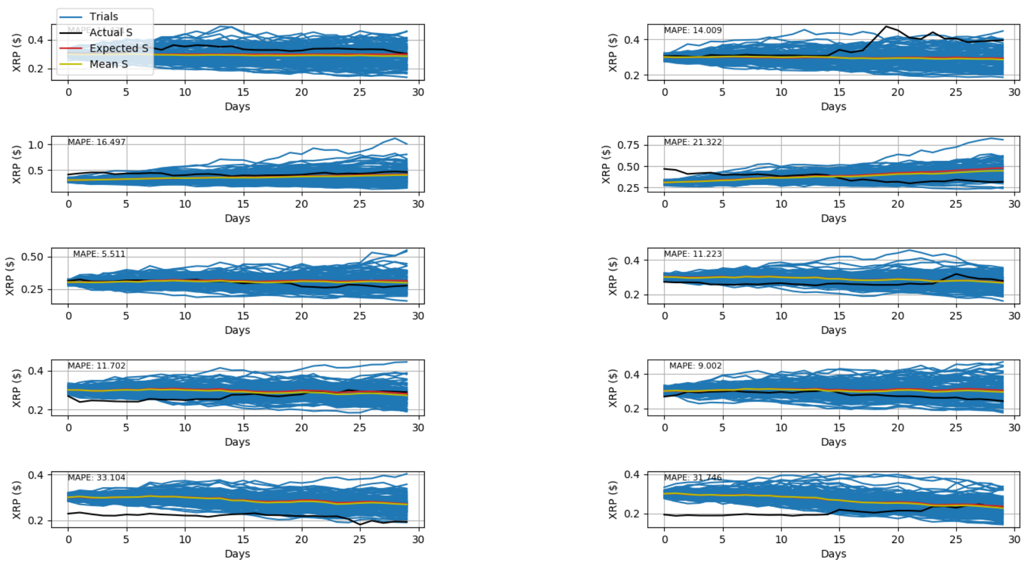


Figure 16: Rolling model simulation result for Ripple with rolling period: 30 days