UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga
(Final project paper)

Povečanje dobička z nakupom in prodajo kriptožetonov na podlagi priporočil iz usposobljenega modela

(Maximizing profit with buying and selling of cryptocurrenices based on recommendations from trained model)

Ime in priimek: Simeon Stanić
Študijski program: Računalništvo in informatika
Mentor: izr. prof. dr. Marko Tkalčič

Koper, oktober 2022

Ključna dokumentacijska informacija

Ime in PRIIMEK: Simeon STANIĆ

Naslov zaključne naloge: Povečanje dobička z nakupom in prodajo kriptožetonov na podlagi priporočil iz usposobljenega modela

Kraj: Koper

Leto: 2022

Število listov: 81 Število slik: 51 Število tabel: 3

Število prilog: 2 Število strani prilog: 26 Število referenc: 10

Mentor: izr. prof. dr. Marko Tkalčić

Ključne besede: podatki, model, kripto-kovanec, dobiček, nakup, lstm, bilstm, gru, rnn, strategija

Izvleček: Trgovanje s kriptovalutami, ki prinaša dobiček, je za ljudi težka naloga, saj je treba upoštevati preveč spremenljivk. Priporočilni mehanizem bo nevronska mreža, usposobljena s preteklimi podatki o kriptovalutah. Predlagani pristop bomo ovrednotili s simulacijo transakcij s kriptovalutami na podlagi preteklih podatkov. Simulirali bomo dnevne nakupe v razponu 600 dni, pri čemer bo v tej simulaciji vsaka metoda napovedovanja seznanjena z vsako strategijo nakupa. Simulacija bo vsebovala 9 najbolj priljubljenih kriptovalut. Rezultat bo vseboval tabelo z vsakim parom (9 parov), da bi ugotovili, kdaj uporabiti kateri par, in jih primerjali, da bi ugotovili, kateri delujejo najbolje. V tej študiji bomo ocenili dve stvari: uspešnost napovedovanja prediktivnih modelov, merjeno v MAE, in rezultate, ki jih bo v simulaciji prinesla metoda napovedovanja v kombinaciji s strategijo nakupa, ki bodo merjeni v sestavljenih obrestih.

Key words documentation

Name and SURNAME: Simeon STANIĆ

Title of final project paper: Maximizing profit with buying and selling of cryptocurrenices based on recommendations from trained model

Place: Koper

Year: 2022

Number of pages: 81 Number of figures: 51 Number of tables: 3

Number of appendices: 2 Number of appendix pages: 26 Number of references: 10

Mentor: Assoc. Prof. Marko Tkalčić, PhD

Keywords: data, model, coin, profit, buying, lstm, bilstm, gru, rnn, strategy

Abstract: Trading with cryptocurrencies so that it yields profit is a difficult task for people as there are too many variables to consider. The recommendation engine will be a neural network trained with past data of crypto currencies. We will evaluate the proposed approach by simulating crypto transactions on historical data. Simulation has be done by simulating daily buying in range of 600 days, in this simulation every prediction method will be paired with every buying strategy. The simulation will contain 9 most popular cryptocurrencies. The result will contain table with every pair (9 pairs), to find out when to utilize which pair and comparing them to find out which ones work the best. In this study we will evaluate two things: prediction performance of prediction methods, measured in MAE, and results yielded in the simulation by prediction method/buying strategy which will be measured in compound interest.

Acknowledgement

First and foremost I want to thank my mentor professor Marko Tkalčić for his guidance. I also want to thank my family and close ones for their endless support.

Contents

List of Tables

List of Figures

Appendices

List of Abbreviations

LSTM	Long short-term memory
GRU	Gated recurrent unit
BILSTM	Bidirectional Long short-term memory
Coin	Cryptocurrency
RNN	Reccurent Neural Network
PM	Prediction Method
TS	Trading Strategy

1 Problem definition

Cryptocurrencies have since changed the world's financial environment by a significant margin. They have become one of the go-to investments by some risk takers and bigger corporations. With the market cap reaching at least 10 billion USD to 500 billion USD for most popular coins, cryptocurrencies can be a good investment depending on certain conditions. It is important to note that the market is highly volatile making it highly unstable. This means, the drop or jump in their prices are somewhat unpredictable which could mean higher rewards for greater risks.

Yielding profit while trading cryptocurrencies is a much more difficult task compared to the regular stock trading due to extra several factors. Fact is that the crypto markets are highly unpredictable, and every world event can have impact on it. Factors that contribute to crypto volatility are that they aren't backed up by anything, so their values solely depend on traders, which means that they have no intrinsic value, unlike stocks. This means that predicting and yielding profit from cryptocurrencies values might be more difficult than stocks.

In this study we want to address the gap which exists in most of the studies regarding crypto predictions, the gap being lack of having simulation environment to truly test the prediction methods and they way we can use their recommendations. Specifically, we will use a recommendation engine coupled with a recurrent neural network (RNN) trained with previous data involving crypto currencies and their closing prices. We intend to evaluate this proposed approach by running simulations crypto transactions based on historical data to find the optimal way of using prediction methods and their recommendations.

2 Related research

There has been numerous research in regular trading (stocks). In the research by Roondiwala et al. they used deep learning techniques to enhance momentum trading strategies in stocks. They used LSTM to predict stock prices, and they divided their research in 4 step which is collecting data, preprocessing, feature extraction and training/testing the model. They have found out that it is possible to predict the stocks fairly accurately with LSTM, also they have shown us how number of epochs affect the accuracy of the model. This study has given us insight how we can utilize the RNNs [9].

There is a study by Kosc et al. on strength of momentum of the value of coin and contrarian effects. What this study shows us is that investment strategies employing momentum and contrarian effects(MCE) achieve abnormal risk-weighted rates of return in comparison to the S&P500 and B&H strategy. Their reasoning for the MCE being so effective is because the crypto market is still young and unstable and the crypto market doesn't have many big investors like stocks and indexes have, so the possibility for monetary gains inside crypto markets are much bigger than in stocks or indexes or any other similar investment options [2].

A study by Bhosale and Mavale that compared volatility of the select coins. They have chosen Bitcoin, Ethereum, Litecoin to research, and these 3 exactly, because they are all different types of coins, based on the technology behind them. This can be quite important for predicting coins value and creating monetary gain from predictions. This shows us, since the coins can be different in many aspects, is that we might need to use different prediction approaches for different types of coins [2].

There is a research by Liu and Tsyvinski that calculates risks and returns of the coins, and they did various statistical studies where they compare factors which correspond to the coin value. The study is quite comprehensive and their most important findings are that most important factors regarding certain coin value is the attention it receives from investors (the investor can be anybody), and momentum of the coin. Fact that a momentum can indicate future value of a coin, is useful because in this study we will use this fact to make decisions trading [7].

A study by Kumar has done a great job comparing multiple different prediction methods, and they showed us the returns and accuracy of their methods. They have compared random forests, ADABOOST, XGBOOST, MLP classifiers. They have not only compared them, but also studied which features have most impact on predictions. This study has shown how they set up the prediction methods. They have found out which among those can be most effective [6].

There is also a study by Abouloula and Krit on comparative automated trading strategies where they have compared the predicting algorithm performances on multiple coins. They have used direct reinforcement learning (DRF) algorithm to predict the future values of coins. This is one of the rare research which actually simulated trading, comparing buying and holding strategy and their DRF performance metrics, with metrics being cumulative returns, sharpe ratio, sortino ratio, maximum drawdown and value-at-risk, this study is unique because of their approach of testing the effectiveness and comparing their algorithm to the baseline of just buying and holding [1].

A study by Hua has compared the performance between the ARIMA and LSTM on predicting bitcoin, in which they explained strengths and weaknesses of both prediction algorithms on bitcoin. What they have done is trained and tested both models on future value of bitcoin, and throughout this method they have determined that LSTM, while it takes considerably more time to train, it is superior to the the ARIMA accuracy wise [4].

A study by Yamak et al. has compared ARIMA, LSTM, and GRU for time series analysis. This study, although they have valid points, they compare their performance only by RMSE and MAPE, this means when we do time series analysis error-oriented metric might not tell the full story how actually good the model is, we seek to prove this statement by implementing simulations which will more or less test the effectiveness of prediction method. They have compared fitting time for models, which is serious factor that needed to be included, but they have used too small data sets, where RNNs are algorithms which require bigger data sets. This study showed us that we needed to use bigger data sets if we plan to implement RNNs [10].

There have been similar research by Ozturk Birim focusing on the comparing recurrent neural network for predicting coin prices. This research focused on performance between GRU, LSTM, BiLST and BiGRU. This research is helpful since it already has elements we want to test in our research. What they have shown is that using RNNs is acceptable and efficient when predicting future value of coins, and that BiLSTM is marginally better than other aforementioned algorithms, but we still want to test rest of algorithms [8].

A study by Koker and Koutmos, presented a model for active trading based on reinforced machine learning and they applied this to five major coins in circulation. Their study is similar to our study because they devised their trading based off machine learning. They showed how can reinforced machine learning can make crypto trading decisions optimize portfolio, which they compared to buy and hold approach. This study showed how can yield superior risk-adjusted returns for investors and even reduce portfolio downside risk across time and if the model is trained adequately [5].

For our contribution to the aforementioned studies, we want to apply previously used techniques and findings to create the two step process where we will test out their real-world applications. Our first step will be comparing Recurrent Neural Networks (RNNs) by predicting various cryptocurrencies (coins), in this step we will test them by Mean Absolute Percentage Error (MAPE). The biggest contribution will come in the second step which is simulating the coin markets based on historical data. In this step we will also introduce trading strategies (TS). There are no studies involving trading strategies in either crypto or stock market (to our current knowledge). Which means we will also simulate various approaches of using the prediction data in the goal of creating monetary gain. After completing this step, we will get result data for each Prediction Method (PM) and TS pair. What this will tell us is the effectiveness of the pairs in real-world application. Few studies have done this simulation approach, but none with RNNs, and by doing this we will find out how truly effective the PMs are and how to use their data in real-world applications, instead of just comparing MAPE or Root-mean-square error (RMSE) between PMs. Aside from effectiveness of PMs we will also compare how difficult are certain coins to predict, and which PMs work best on what coins. This is important since not every coin has the same technology behind it or same following along with many more factors.

3 Data collection and pre-processing

Data collected in this experiment is taken from Binance's historic data. Scope of historic data collected for 9 cryptocurrencies namely:

- Binance Coin (BNB)
- BitCoin (BTC)
- Ethereum Classic (ETC)
- Ethereum (ETH)
- Chainlink (LINK)
- Litecoin (LTC)
- NEO (NEO)
- TRON (TRX)
- Ripple (XRP)

Data collected from these 9 coins were from the period starting from 15.3.2020 to 15.3.2022. Rows were segregated per day. Data contained: price in USD, date, open price, close price, high, low, trade count, volume USD, volume of coin.

Example of dataframe before normalization:

	unix	date	symbol	open	high	low	close	Volume BTC	Volume USDT	tradecount
52	1.642810e+12	1/22/2022 0:00	BTC/USDT	36426.46	36817.74	33950.00	35043.73	676272.904	2.390697e+10	6139534
51	1.642900e+12	1/23/2022 0:00	BTC/USDT	35043.73	36496.92	34588.80	36230.01	369372.903	1.309740e+10	3421563
50	1.642980e+12	1/24/2022 0:00	BTC/USDT	36230.00	37528.00	32853.83	36648.31	823000.702	2.852382e+10	7379240
49	1.643070e+12	1/25/2022 0:00	BTC/USDT	36648.30	37561.03	35693.00	36934.26	409535.517	1.496747e+10	4251175
48	1.643160e+12	1/26/2022 0:00	BTC/USDT	36934.26	38886.58	36232.42	36787.43	554457.384	2.085887e+10	5467829

Figure 1: Data before normalization.

3.1 Data processing and normalization

This data was then pre-processed first to ensure that we can use RNN algorithms on them. First, we needed to ensure that there were no missing or incomplete data, since data was complete, there was no need for any modifications. Since we decided on using the uni-variate models, we removed every other column except the close price, since we only want to predict the close price of that particular day. Since we used the data set that spanned from 15.3.2020 to 15.3.2022, we had 600 days of data, thus we wanted to do time series analysis on the data. We have decided on 600 days, since it is a good size for avoiding flukes, and it was compact enough so it does not take ridiculous amount of time for RNN algorithms to run it. Then, the data were normalized, so the algorithms can work on the data. To do this, we used min-max values to scale these data thereby normalizing them into 0 to 1 values. This was necessary to do since that is a requirement for the RNN to be able to do time series analysis on the data.

```
array([[0.36792941],
       [0.37178714],
       [0.42376662],
       [0.43777629],
       [0.46013095]])
```

Figure 2: Data after normalization.

4 Predicting the value of cryptocurrency

After collecting and preprocessing the data we built the models to predict the future value of the coins, that is we want to do univariate time series analysis on close price. To predict values we have chosen RNN models. The reason for which we have chosen them is because they are well researched, so we can refer previous studies to help with our research, other the reason is that RNNs don't require feature engineering. Arima is also popular choice for predicting future values, but we chose RNNs because they work better on bigger data sets. Another advantage of using RNNs in our research is because they are easy to implement, but they do require careful hyper-parameter tuning, which is the hardest part about using RNNs.

To predict future value of cryptocurrencies, we have utilized RNNs. The 3 prediction methods are:

- GRU model (Gated recurrent unit)
- LSTM model (Long short-term memory)
- BiLSTM model (Bidirectional LSTM)

4.1 Models

Here we have the models which we have used in our research.

4.1.1 LSTM

According to Ozturk Birim [8], the Earlier forms of RNNs *have weakness of not capturing long-term temporal dependencies of sequential data. LSTM was emerged as a type of RNN to overcome this weakness. LSTM networks are effective in handling long-term temporal dependencies in sequential data while do not experience the optimization difficulties which the traditional RNNs face (Greff et al., 2017). An LSTM cell has three gates which are input, output and forget gates. LSTM also has a cell state which stores the temporal state and control the gates (Shahid et al., 2020). Forget gate determines* *the amount of the information which should be thrown away from the cell state. Input gate regulates the amount of new information stored in the cell state while the output gate determines the information that should be sent as the output from the cell state (Ayoobi et al., 2021).*

4.1.2 BiLSTM

Again according to Ozturk Birim [8], Bi-directional application *is a different way of stacking RNNs. Bidirectional approach to RNN was first introduced by Schuster and Paliwal (1997). Bi-RNN not only uses past information while training but also utilizes future information of a specific time frame. This is handled by splitting neurons of a RNN into two parts; one is responsible for future states, the other part is responsible for past states (Schuster & Paliwal, 1997). Each group of neurons constructs the forward and backward layers, and these two layers are linked to each other. Various merging methods are used to combine the outputs of the forward and backward layers to produce a final result (Althelaya et al., 2018). When the two layers of LSTM one of which flows in forward direction and the other one flows in backward direction are stacked together, a bidirectional LSTM (Bi-LSTM) structure is formed. Ayoobi et al. (2021), Hamayel and Owda (2021) and Shahid et al. (2020) applied Bi-LSTM to the time series data and indicated the powerful functionality of LSTM.*

4.1.3 GRU

By definition of Ozturk Birim [8], a GRU cell *includes two gates as opposed to an LSTM cell. These two gates are an update gate and a reset gate. The weights of these gates are adaptively updated in the learning phase of the algorithm (Dey & Salem, 2017). GRU has achieved remarkable success on various tasks including computer vision and sequential data. The success of GRU lies in the simpler structure than LSTM with two gates instead of three (Yang et al., 2020). can be regarded as a variant of LSTM. The aim of the GRU is the same as LSTM that is to provide improvement of a traditional RNN in predicting sequential data. GRU can even be more successful and faster than LSTM in low dimensional sequential data due to its smaller number of gates and parameters. Additionally, training in LSTM can be longer since the network structure is more complex while LSTM can get better results in high dimensional data (Lai et al., 2021).*

4.1.4 Tuning the models

We have used code by Follonier as a basis for our code for RNN models [3]. We have used this code to get baseline results, and then we changed the parameters to achieve lowest MAPE and highest execution speed. We could have aimed simply for lowest MAPE possible, but the execution speed was a factor since there was high amount of data.

All 3 models when training use batch size of 16, and 100 epochs and 13 neurons on input layer.

We have used 100 epochs because it is a good number which doesn't either over-train or under-train the models and training time is reasonable which is also a factor.

Number of neurons in input layer is 13, because there are 13 parameters which are taken in as a input.

We have chosen batch size 16 instead of 13, even though final batch has fewer samples to work through, the performance improved slightly with very little cost to processing time.

The GRU model that is set up using one GRU layer and two dense layers.

GRU models graph are depicted in Fig. 3.

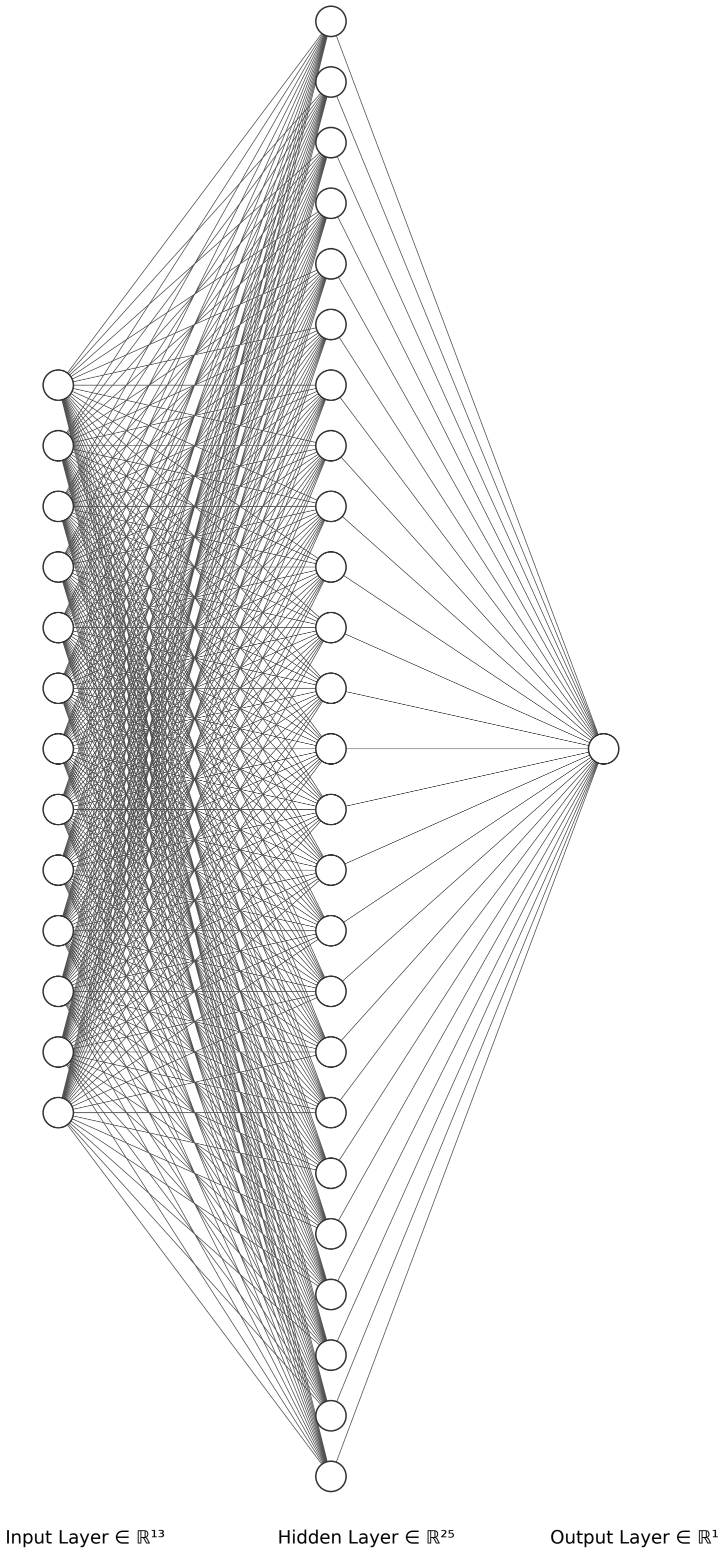


Figure 3: GRU model architecture.

The LSTM model is set up using two LSTM layers and two dense layers.
The BiLSTM model is set up using two BiLSTM layers and two dense layers.
BILSTM and LSTM models graphs are depicted in Fig. 4.

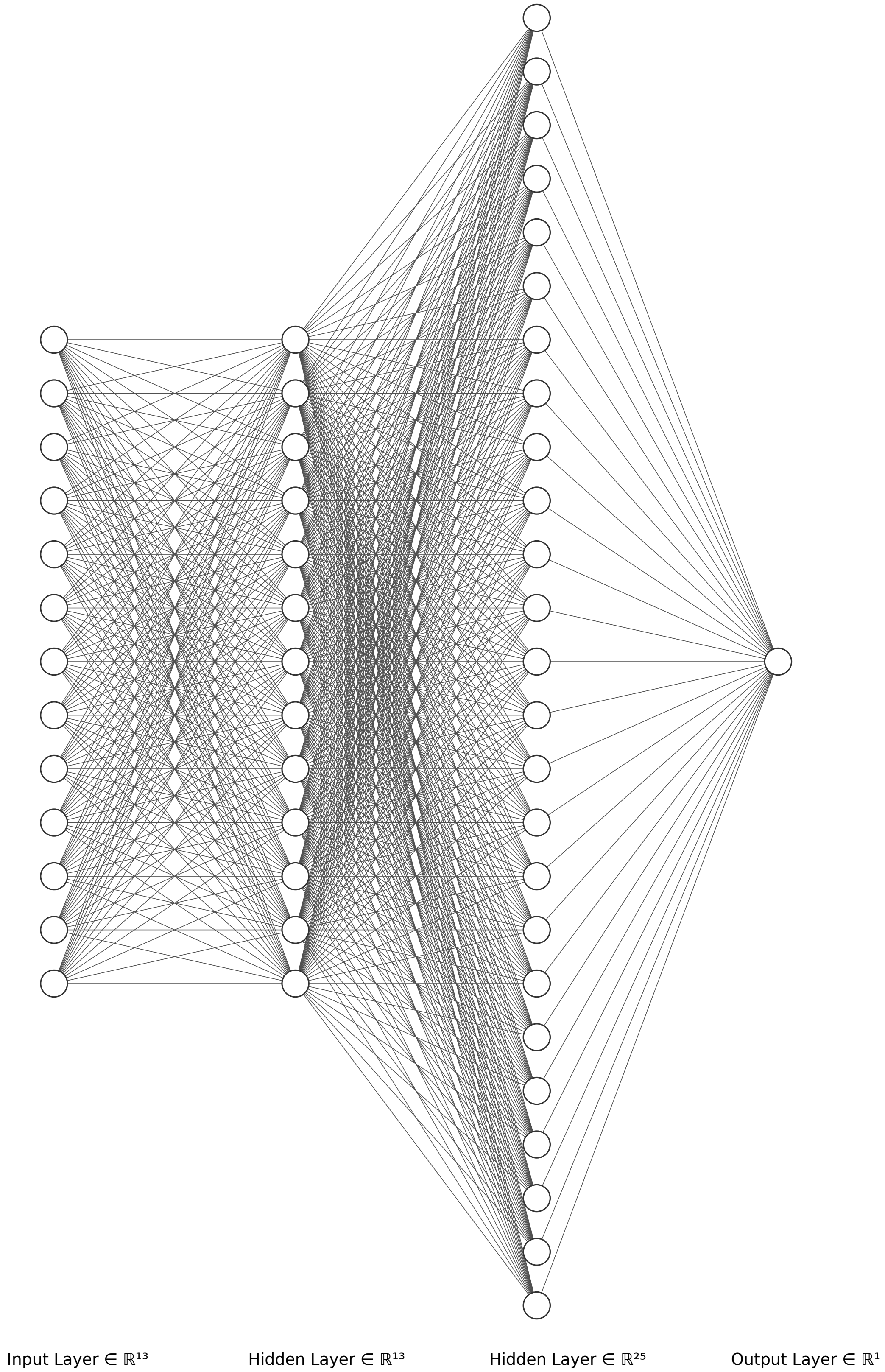


Figure 4: LSTM and BiLSTM models architecture.

4.1.5 Fitting the models

Now that we introduced the models, we want to explain the way models are fitted for forecasting the prices. The models are built to be uni-variate, meaning models take into consideration only the past data of the close price. Initial idea contained more variables, but we opted to use a uni-variate approach because this reduces the chances of over-fitting.

To validate this, we use data points that were 6 days ahead of the training model. This is so we can be more flexible with the buying strategies that take into account further predictions beyond just one day. Next, we determine the appropriate set of attributes that work for a window length of 6 days. This produces a prediction on the 19th day from the first data point. We computed the mean absolute percentage error (MAPE) to determine the best description of the data for forecasting the prices. We learned that we needed to have 13 features for predicting the price for a given day (in one row). This is from having total time-series data of 152 rows for training. We do this to minimize the MAPE without over-fitting the models at the same time.

From this we have used 145 days as training set and 6 days as a testing set.

These parameters are results of hyper-parametering, which means we needed to tune the models by testing different data point lengths. This meant that we had to rerun the model multiple times to determine optimal parameter lengths, and optimality was judged by MAPE and execution time of the model. Execution time was a factor because we had 3 models, 9 coins and 100 runs per pair, so we needed to fit models 3x9x100 times to for the study, and this took immense amount of time to get predictions required for the study.

4.1.6 Testing the models

In table 1, you can see the MAPE values averaged from 1% to 6.5%. In most cases it ranged between 2.5% to 4%. This is average for all coins, but some coins as you can see in table 1 have varied MAPE, this discovery told us that some coins are less predictable than others. We mentioned in related research that there are coins which are more volatile and harder to predict than others, which proves this notion. Also, there is a fact that there are different types of coins based on technology behind them, so we can say that different coins need different prediction approach, unlike the universal approach we used for every coin so far. Also we can see in figures 6, 5, 7 the differences in prediction algorithms.

The overall MAPE was similar for all 3 models during the pre-war and post-war dates where there is almost no difference between models. The BiLSTM achieved the highest accuracy in terms of its MAPE. However, it is important to note that MAPE alone is not a sole indicator of the success of these models. To verify this, we compare their actual performance on the prediction of crypto prices from our simulations.

Table 1: MAPE average with PM x coin pairs over 600 days.

Coin/Model	LSTM	BILSTM	GRU
BNB	4.022	3.417	6.489
BTC	1.31	1.015	2.221
ETC	3.722	2.653	5.102
ETH	2.522	2.832	4.02
LINK	1.708	2.088	3.794
LTC	1.491	1.005	2.7447
NEO	2.25	1.426	3.352
TRX	2.006	1.483	2.782
XRP	4.273	3.073	4.315

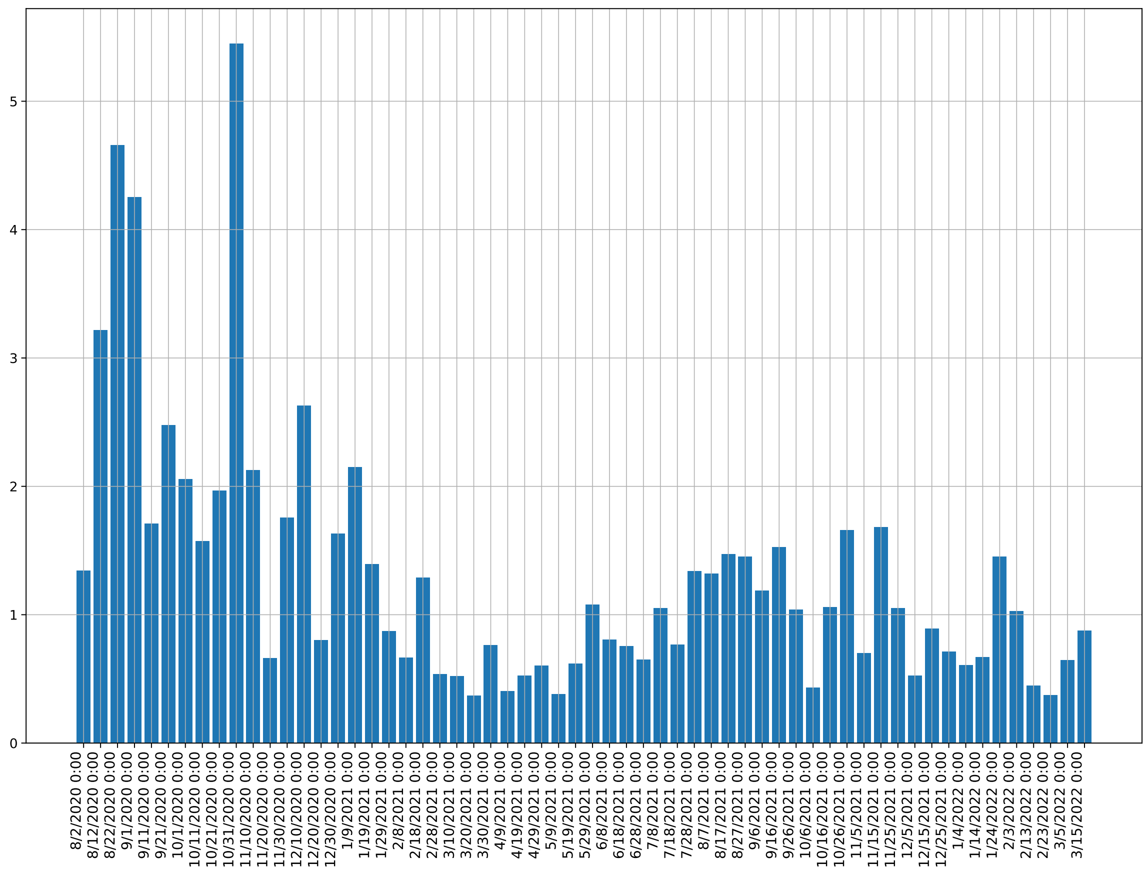


Figure 5: MAPE average on LSTM while predicting bitcoin.

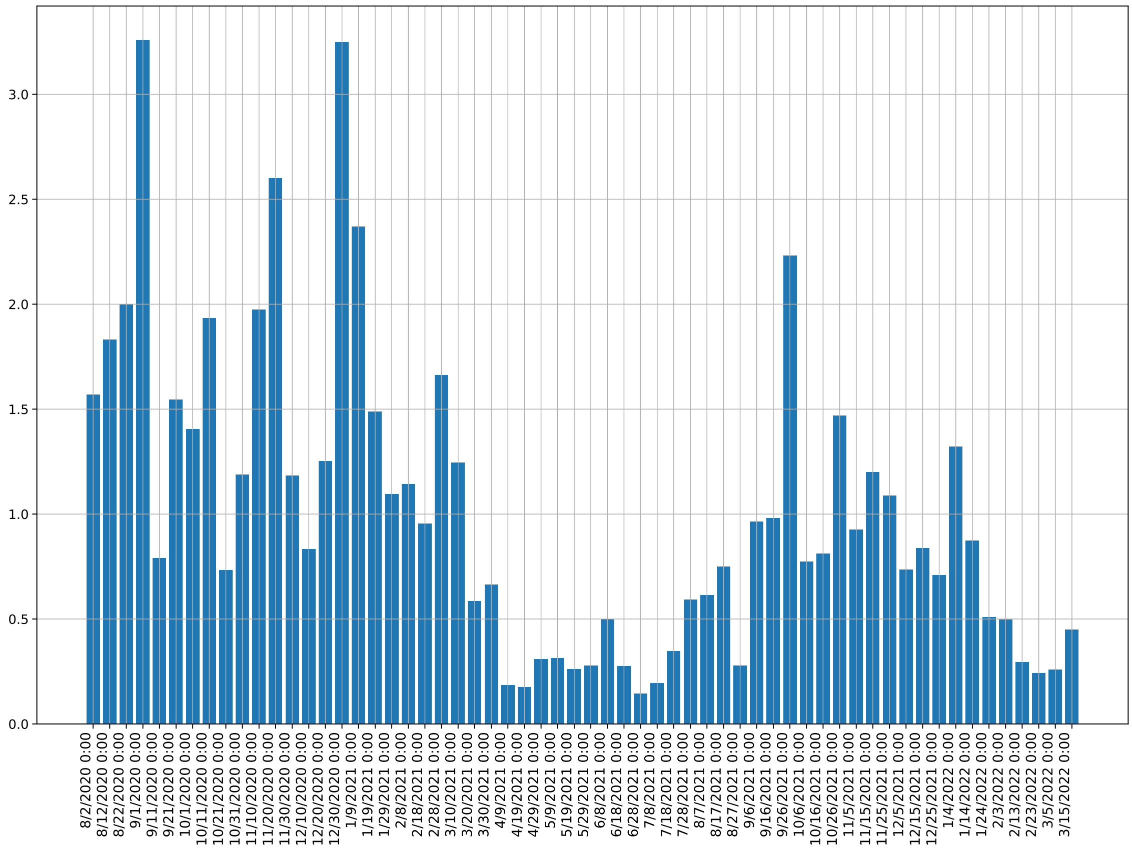


Figure 6: MAPE average on BILSTM while predicting bitcoin.

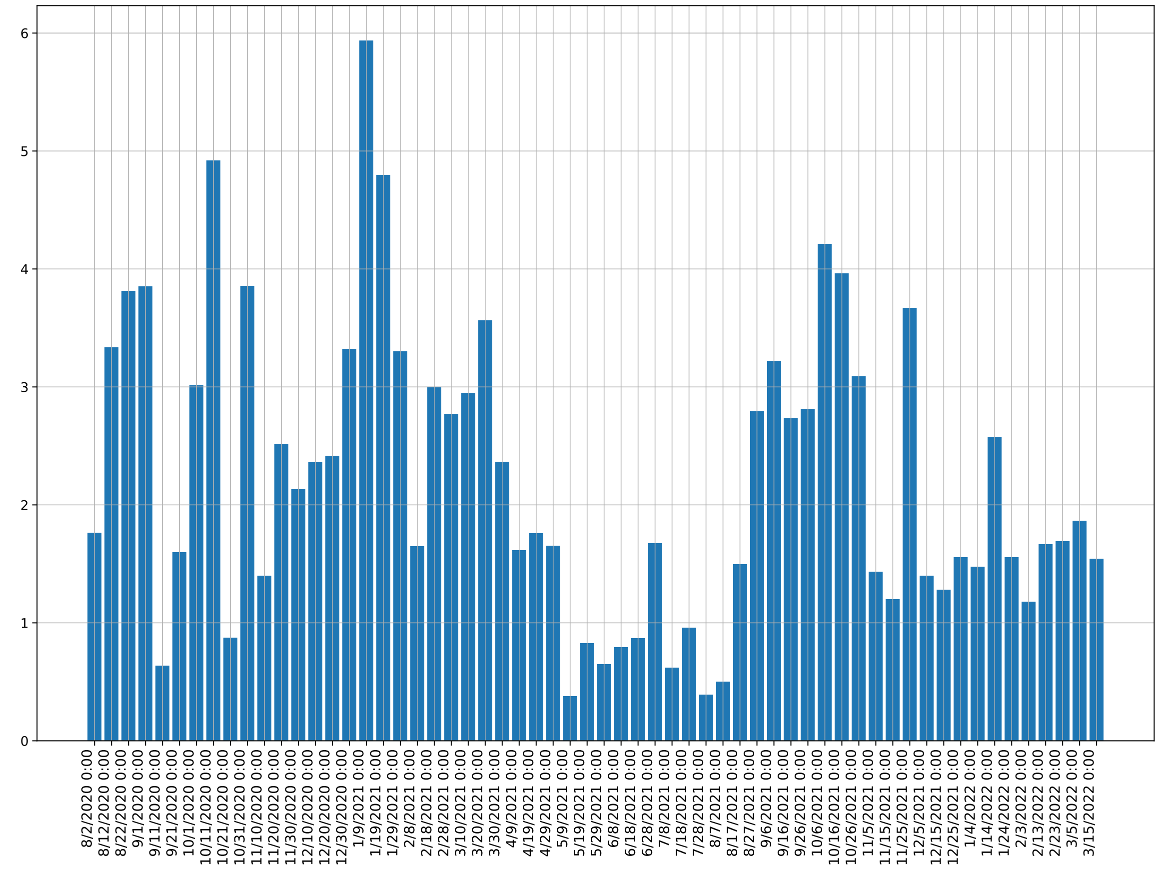


Figure 7: MAPE average on GRU while predicting bitcoin.

4.2 Predicting the values

For this research we have predicted 600 days ranging from 7/24/2020 to 3/15/2022 for predicting a single coin. For each 6 days model was retrained again to accommodate for new variables in the data. This means that each model was fitted 100 times for predicting single coin in this time interval.

We got 27 data-sets with predicted values for every coin x prediction method pair.

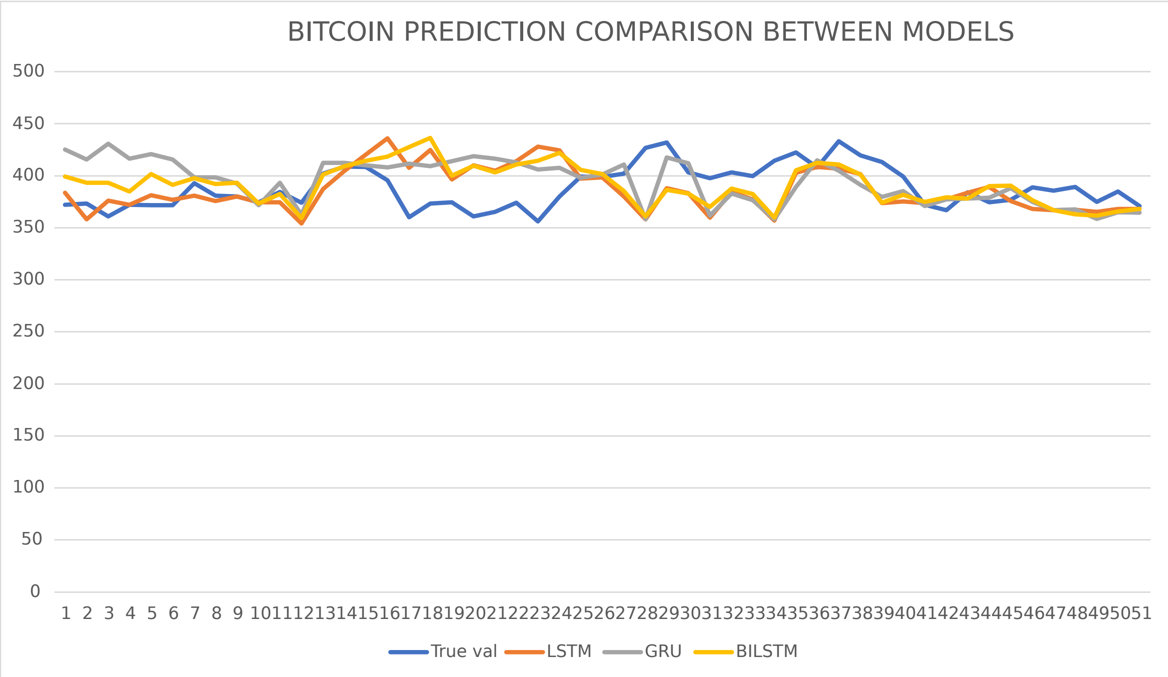


Figure 8: Graph of prediction comparison between 3 algorithms.

5 Trading strategies and simulation

In this chapter we will define what trading strategies are and what simulation does. The goal of this is to utilize the recommendations given by prediction methods, to find how they would perform in real life, and give insight how we can improve them.

5.1 Trading strategies

We have devised multiple buying strategies for buying cryptocurrencies:

- Waiting out the dip (on a single cryptocurrency) and buying at bottom (buying loop)
- Transfer by detecting higher momentum in some other CC
- Equal spread on currently n highest predicted growing CCs

5.1.1 Waiting out the dip (WD)

This method is based on buying strategy used in classic crypto trading, and it is similar to buying low selling high principle in business. We have taken that as a basis of this strategy. The strategy is implemented like this: Take predicted array (array of predicted values over next n days), and make an average of predicted values in next n days. Then compare current day price compared to the averaged out value, if the average for next n days is higher by at least 3 percent than the current day price then keep/buy the coin, depending if it's already bought or not. If it's the opposite case where the current price is higher than the averaged value +3% then the strategy gives signal to sell/not buy the coin. If it's bought then immediately sell. After simulating, we keep data whether we owned coin at that time as a binary array, with each entry representing n time step, by 1 we say we have coin in that particular time step, and by 0 we say we don't have coin. Here we have algorithm for this strategy as it was written in python.

```
def WDstrategy1(TruMat,PredMat,coin):

    total=0;
    average=0;
    registry=[]
    pred=PredMat[:,coin]
    truarr=TruMat[:,coin]
    for i in range(firstpoint,datalen,skip):
        predictedarray=pred[i:i+skip]
        lastval=truarr[i-1]
        total= sum(predictedarray)
        average=total/len(predictedarray)
        if( (average/lastval) > 1.03 ):
            registry.append(1)
        else:
            registry.append(0)

    return registry
```

5.1.2 Momentum transfer (TR)

This method is based on classical trading approach called momentum trading. It is fairly simple strategy of trading. The idea of this strategy is to detect which coin is going to grow the highest in next n days, then buy that coin, and sell it when we detect some other higher growing coin. This is executed by taking the predicted matrix in which columns represent coins and rows represent the predicted values of these coins. Then the predicted values are averaged out, and compared to the current value. Then whichever coin has the highest growth is bought and kept. If in next few days there is coin with bigger growth, then the previous one is sold and new one is bought. After simulating, we keep data whether we owned coin at that time as a binary matrix, with each entry in a row representing n time step, and every column representing the different coin, by 1 we say we have coin in that particular time step, and by 0 we say we don't have coin.

```
def TRstrategy1(TruMat,PredMat):

    registrymatrix=[0]
```

```
    for i in range(firstpoint ,datalen ,skip):
        last_value = truthMatrix[i-1,:]
        predictedmatrix= PredictedLSTM[i:i+skip ,:]
        totalarray = np.sum(predictedmatrix , axis=0)
        averagearray=totalarray/np.size(predictedmatrix ,0)
        diff=averagearray-last_value
        ratio=diff/averagearray
        indices = np.argsort(-ratio)
        indices= indices[:1]
        registrymatrix.append(indices)

    return registrymatrix
```

What this algorithm returns is the registry matrix which indicates which coin is currently owned, signified by binary system.

5.1.3 Equal spread (ES)

This strategy is also based on momentum trading and previous strategy of detecting momentum. The only difference between previous strategy and this one is that in previous strategy the simulator takes into consideration only one highest growing coin, but in this we specify the n number of coins we want to have at all times. Essentially what this strategy does is looking at n highest growing coins and owning them as long as they are high growing, then selling them if there are more favourable coins as growth goes. The here important thing is to determine the n that works best in most cases. Throughout testing we have determined that best number for n is 3 for our application (keep in mind that we predicted only 9 coins so n might vary greatly with different number of coins that we take into consideration). After simulating, we keep data whether we owned coin at that time as a binary matrix, with each entry in a row representing n time step, and every column representing the different coin, by 1 we say we have coin in that particular time step, and by 0 we say we don't have coin.

```
def ESstrategy1(TruMat,PredMat,n):

    registrymatrix=[];
    for i in range(firstpoint ,datalen ,skip):
        last_value = truthMatrix[i-1,:]
        predictedmatrix= PredictedLSTM[i:i+skip ,:]
        totalarray = np.sum(predictedmatrix , axis=0)
```

```
        averagearray=totalarray/np.size(predictedmatrix,0)
        diff=averagearray-last_value
        ratio=diff/averagearray
        indices = np.argsort(-ratio)
        indices= indices[:n]
        registrymatrix.append(indices)

    return registrymatrix
```

This algorithm varies from the momentum transfer only in number of coins, the principle is the same.

5.2 Simulation

The goal of simulation is to simulate real life crypto trading environment. The simulation is done on time period from 7/24/2020 to 3/15/2022. To achieve this, we use true data and predicted data from this time period, plug it into strategies and simply get buy sell signals, and based on those signals we compute the profit/loss from true data.

Here we have the simulator that we used to test the efficiency of the prediction models/buying strategies pairs. The simulator takes as an input three parameters: predicted data, true past data which is +1 length of predicted data, and how often the trading process is rerun - we put an integer which represents the time skip in days in simulation (time skips represent what time interval we take predictions and make calculations and iteration of simulation algorithm) and this number also represents length of the prediction window that we take, since we don't need predictions beyond the time interval we operate in.

The simulator operates by taking time skip and length of data as inputs, then it calls TS functions, which then gets back the registry on the state of the coins. Meaning we get back an array with the following dimensions: number of coins x days/time skip length. In this array we have 2 states of coin, 1 indicates that we own some value in that cc, and 0 indicates we don't. When we get the registry back, we take the historic data and coin registry, and we compute the profit in percentages over time over every coin (how we compute the profit in the end depends on the buying strategy).

5.3 Results

We used RNN to predict values of coins from 7/24/2020 to 3/15/2022, meaning we have predicted 600 days, and we have used historic data to verify the results.

Here are baseline results in which the coins are purchased on 7/24/2020 and their values are checked on 3/15/2022.

To find baseline results we used a strategy in trading assets which is called buy and hold, and is used as a baseline in many other research.

Table 2: Holding profit over 600 days by Buy and Hold strategy.

Coin	Holding Profit
BNB	$+1,884\%$
BTC	+307%
ETC	+284%
ETH	+744%
LINK	+75%
LTC	+138%
NEO	+67%
TRX	+218%
XRP	+258%

5.3.1 WD strategy

In table 3 we have coins/model percentages of profit achieved in 600 days simulation

Table 3: WD Strategy result on Coin x Prediction Method pairs by 600 days of simulation.

Coin/Model	LSTM	BILSTM	GRU
BNB	2001.041	2001.041	2001.041
BTC	521.905	587.691	511.614
ETC	184.504	184.504	184.504
ETH	665.077	665.077	656.654
LINK	461.252	470.397	486.033
LTC	649.705	548.211	629.062
NEO	477.027	456.689	464.349
TRX	642.626	642.626	642.626
XRP	439.011	461.282	472.442

The remainder of the results are in appendix A.

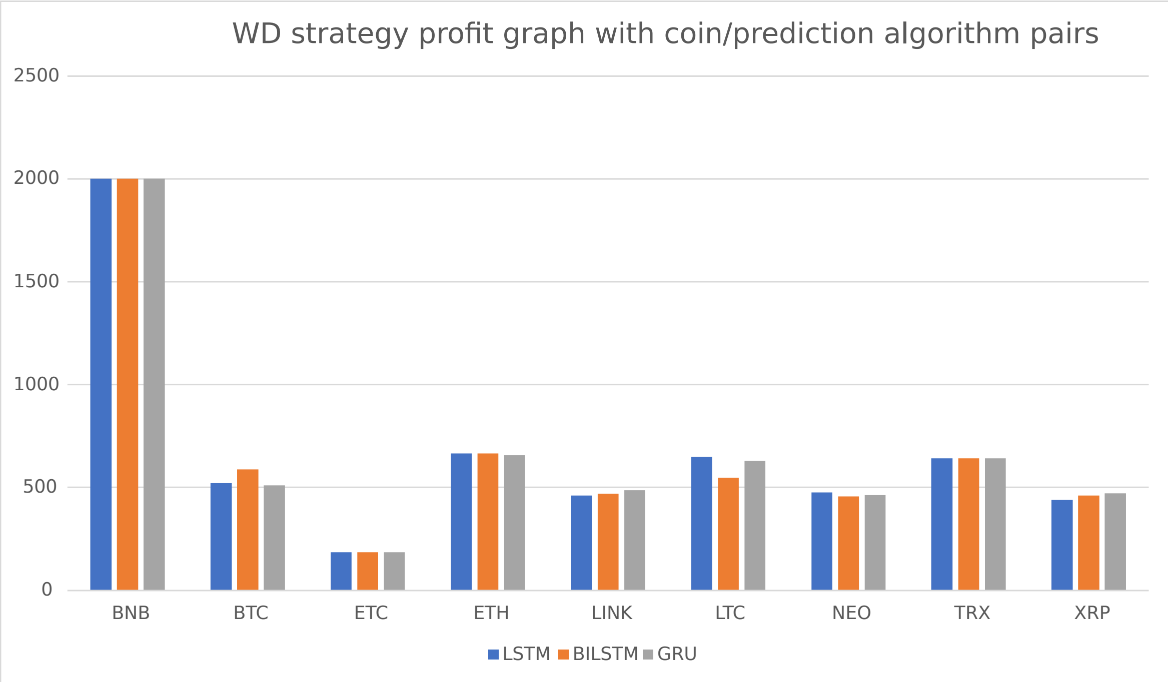


Figure 9: Bar plot of WD profit table3 .

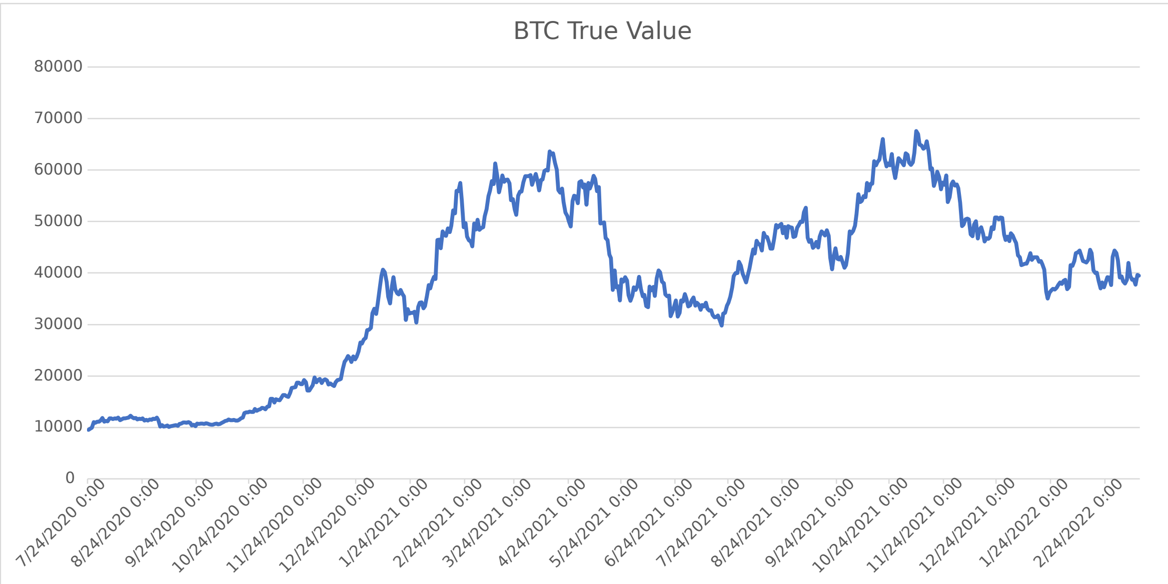


Figure 10: BTC Value over last 600 days.

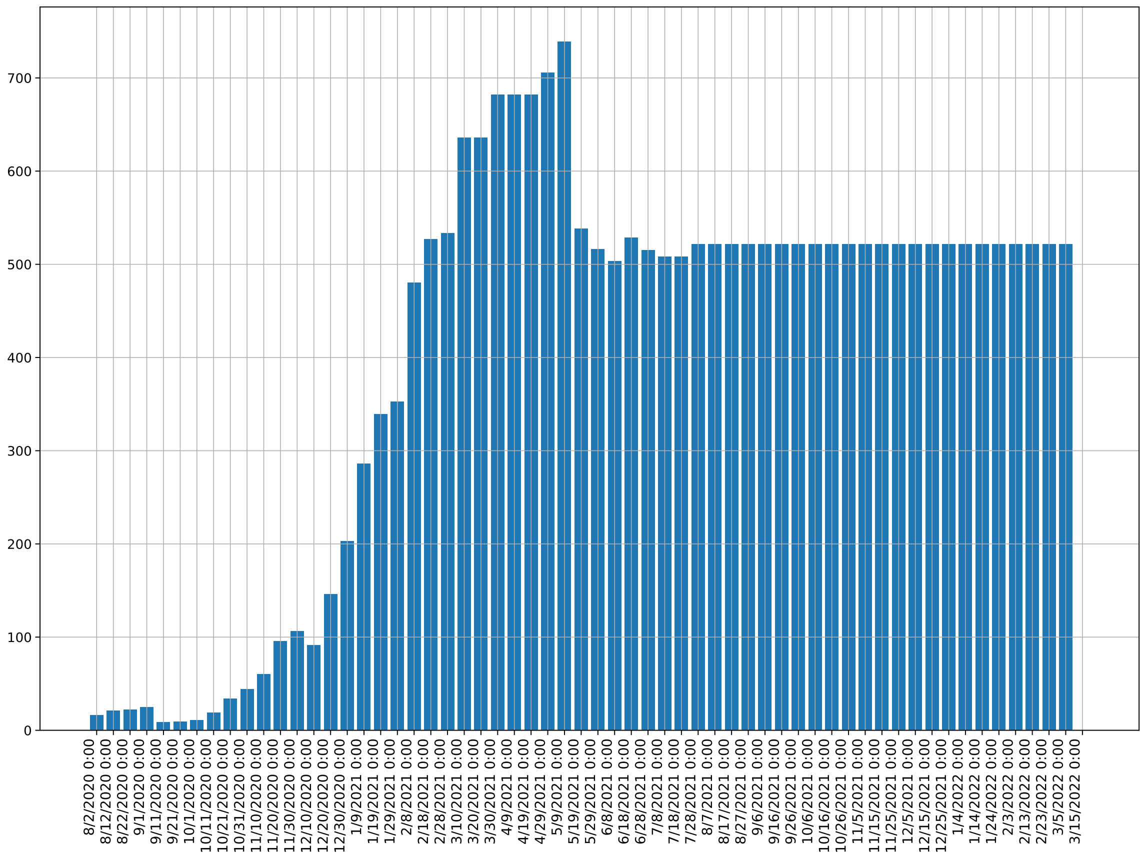


Figure 11: BTC results in profit percentage while using LSTM prediction method.

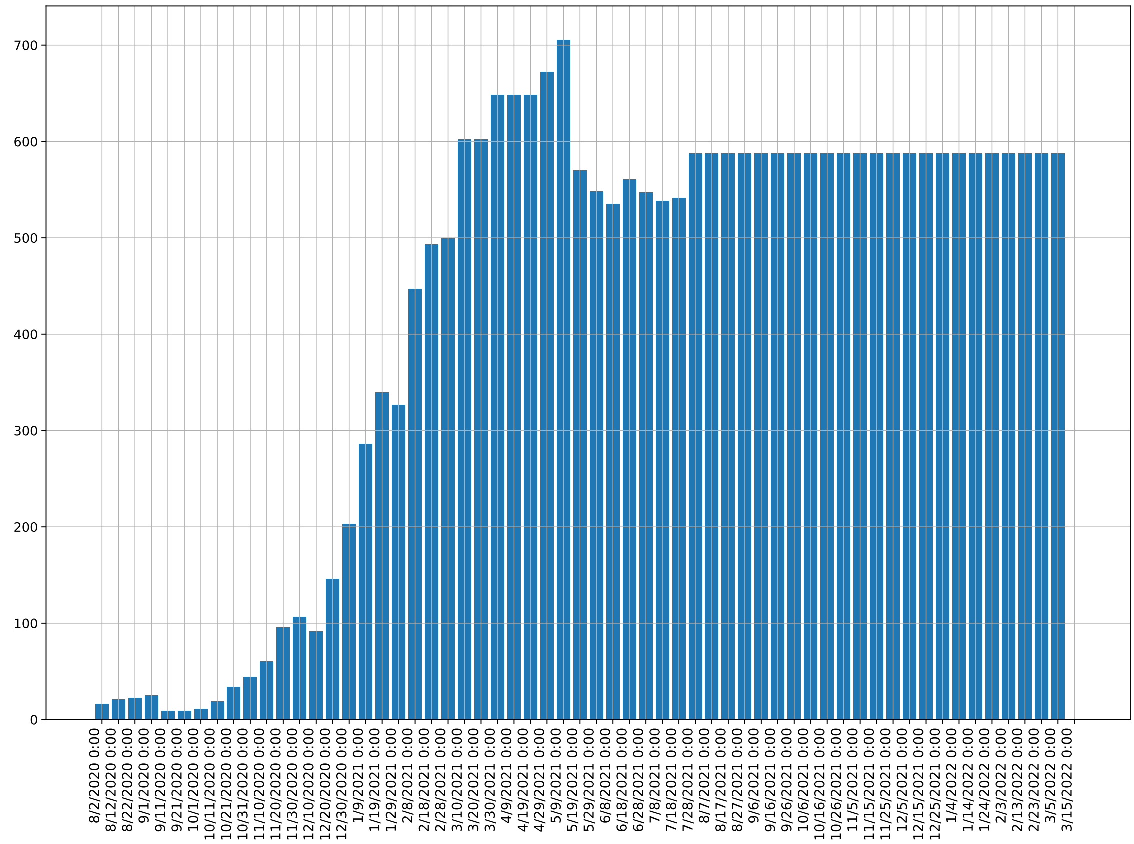


Figure 12: BTC results in profit percentage while using BILSTM prediction method.

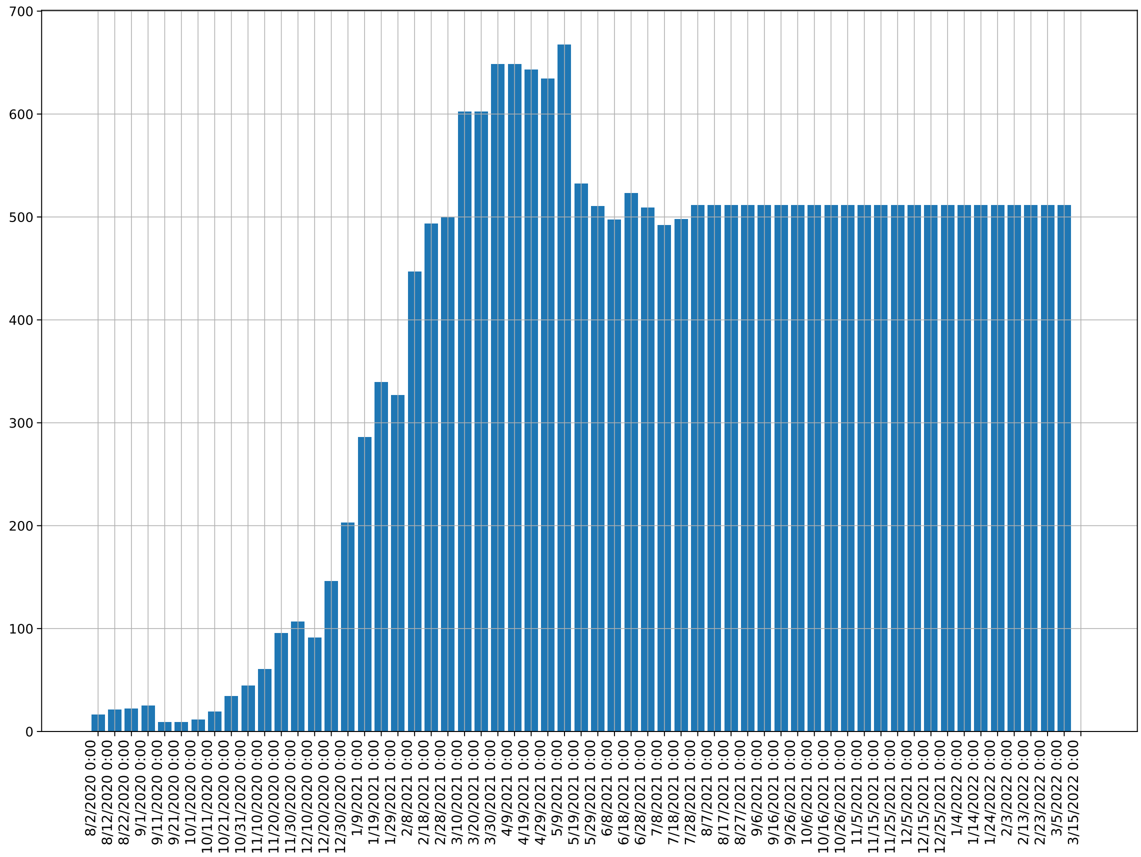


Figure 13: BTC results in profit percentage while using GRU prediction method.

5.3.2 Momentum transfer (TR) strategy

This strategy, which chooses what single coin to own at which time interval based on momentum calculated from predicted values, has percentage of profit of 331.737% across all prediction methods. Initially, for this strategy it was expected to have varying results based on prediction method. The reason why the results are the same is that all methods are equally good at predicting the momentum. The reason why this strategy hasn't scored higher in terms of profit is that it never held BNB, since the prediction method haven't recognized it as a highest growing, even though the BNB grew the most out of all coins.

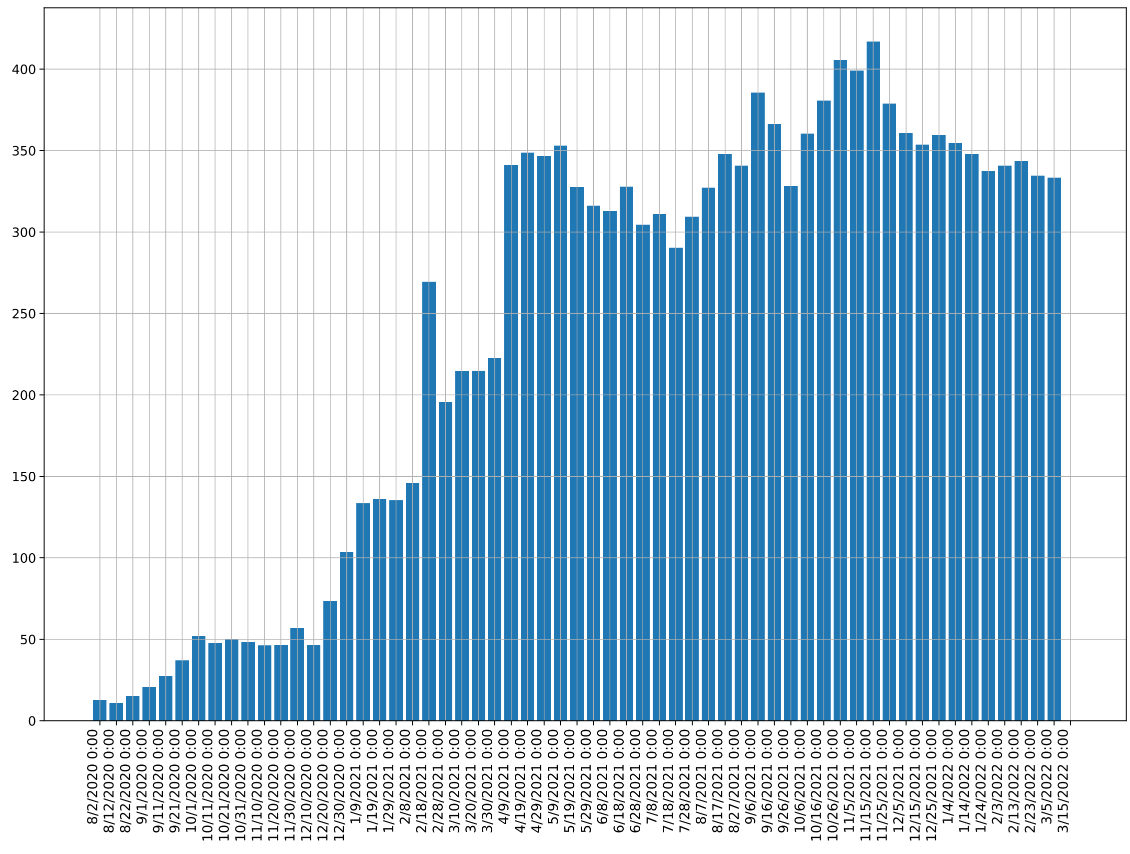


Figure 14: TR strategy results.

As we can see in this figure 14, the growth is consistent, while loses are minimal.

5.3.3 Equal spread (ES) strategy

For this trading strategy, all 3 prediction methods have achieved same results. We need to mention that these results might be more optimistic than usual results, reason is that we have used BNB as one of the coins, which grew +1,884%, so we can see, the more coins we own the profit percentage is lower. Since we spread money equally over n highest growing coins, the profit dipped since there was less money on highest growing coin. We also did simulation where we have excluded BNB from testing, you can find these results in appendix B.

This strategy is similar to the TR strategy, and only difference is that we include n number of coins, so this strategy also gets same results across all PMs.

In this strategy, the most important parameter is number of coins owned at same time, this means that at same time we can have n number of coins. This strategy has the same principle as the TR strategy, but we can own n number of coins which have most momentum at same time.

Nine coins are the baseline, so comparing to just holding all 9 coins we have achieved more profit by shifting the momentum, with owning 3 coins at same time gave the best results.

Here we have results per coins owned at same time interval:

- Two coins: 1109.794%
- Three coins: 1190.910%
- Four coins: 891.018%
- Five coins: 642.305%
- Six coins: 654.265%
- Seven coins: 675.953%
- Eight coins: 400.058%
- Buy and Hold all 9 : 441.420%

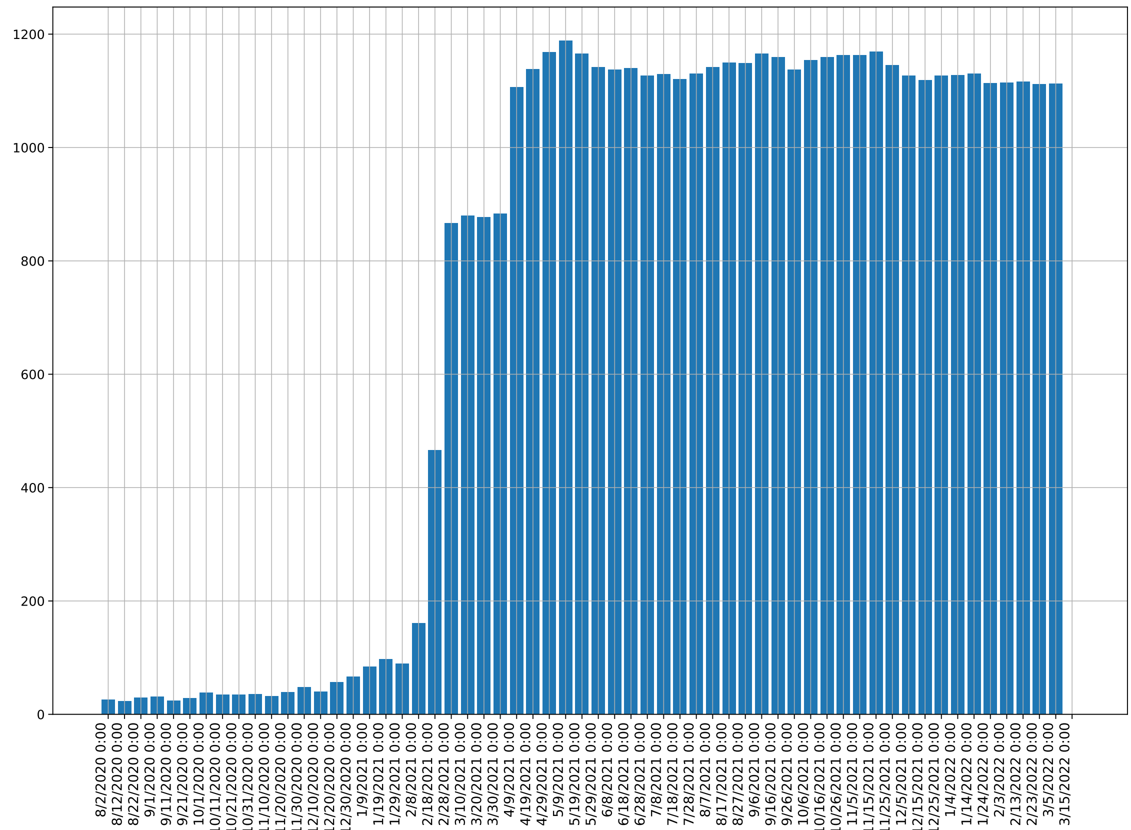


Figure 15: Equal spread result with 2 coins.

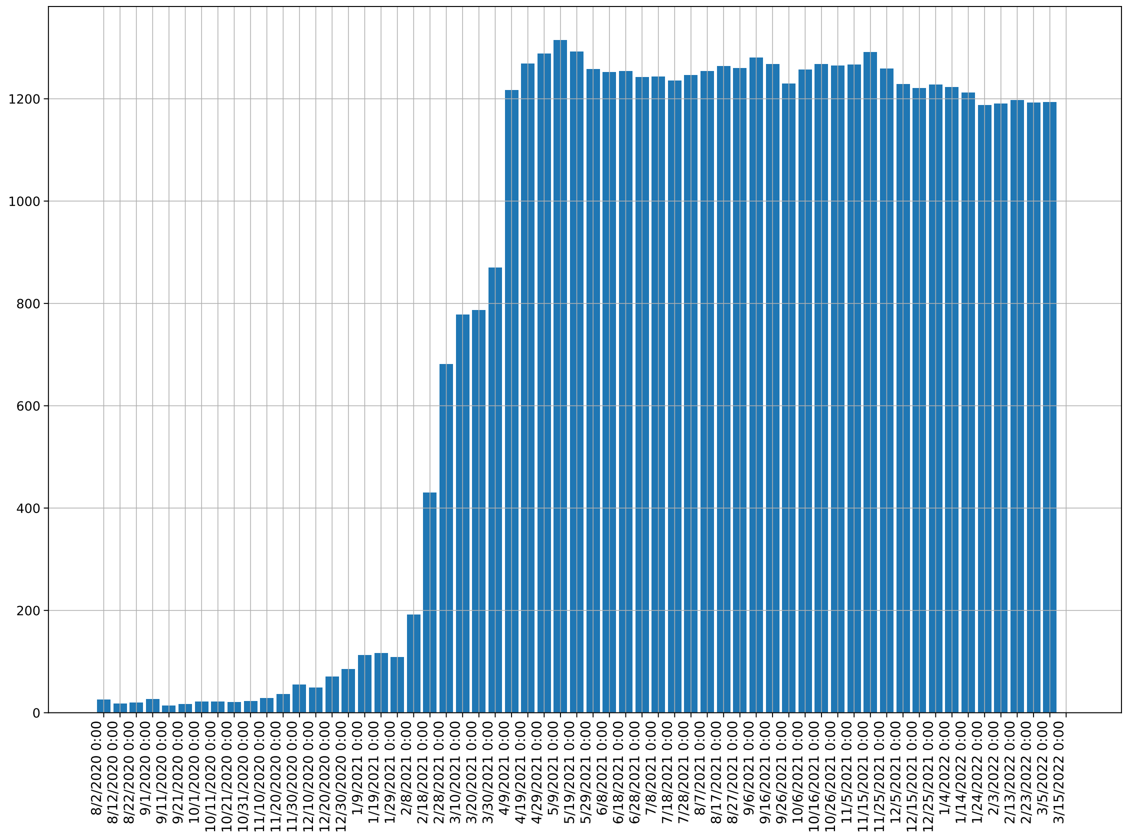


Figure 16: Equal spread result with 3 coins.

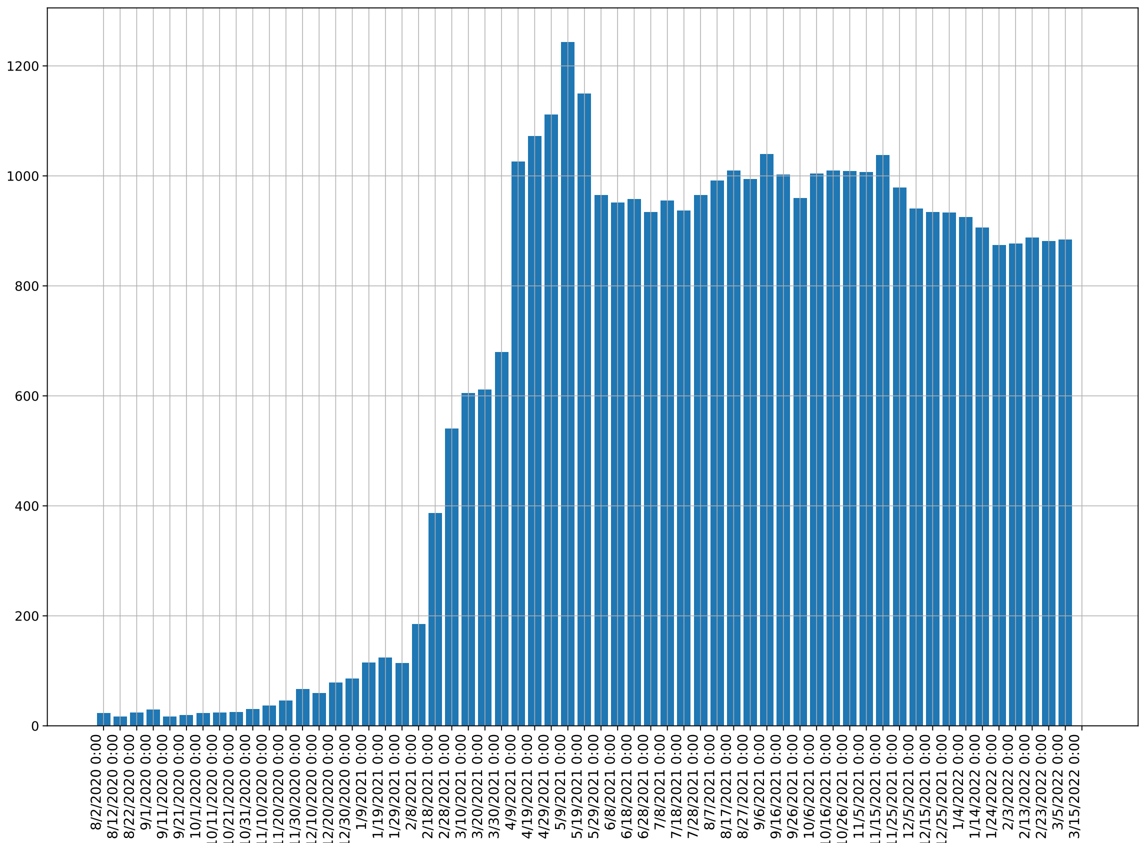


Figure 17: Equal spread result with 4 coins.

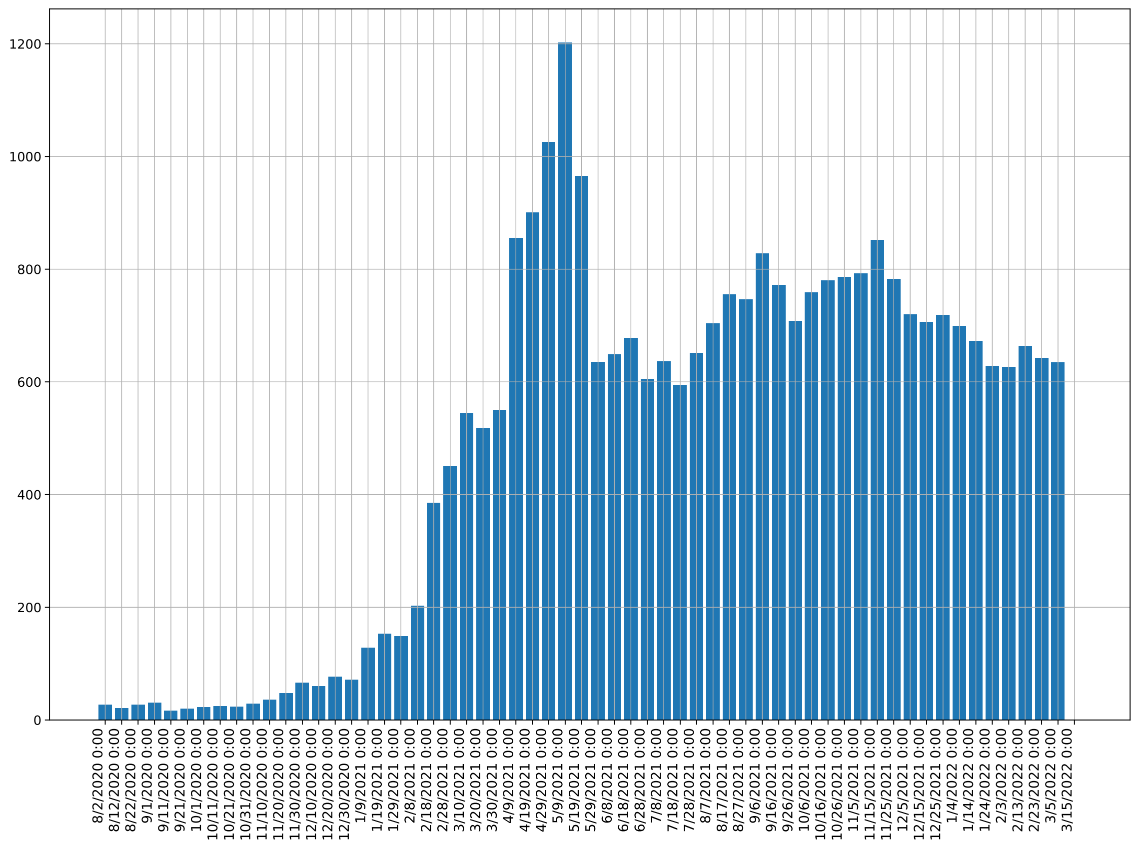


Figure 18: Equal spread result with 5 coins.

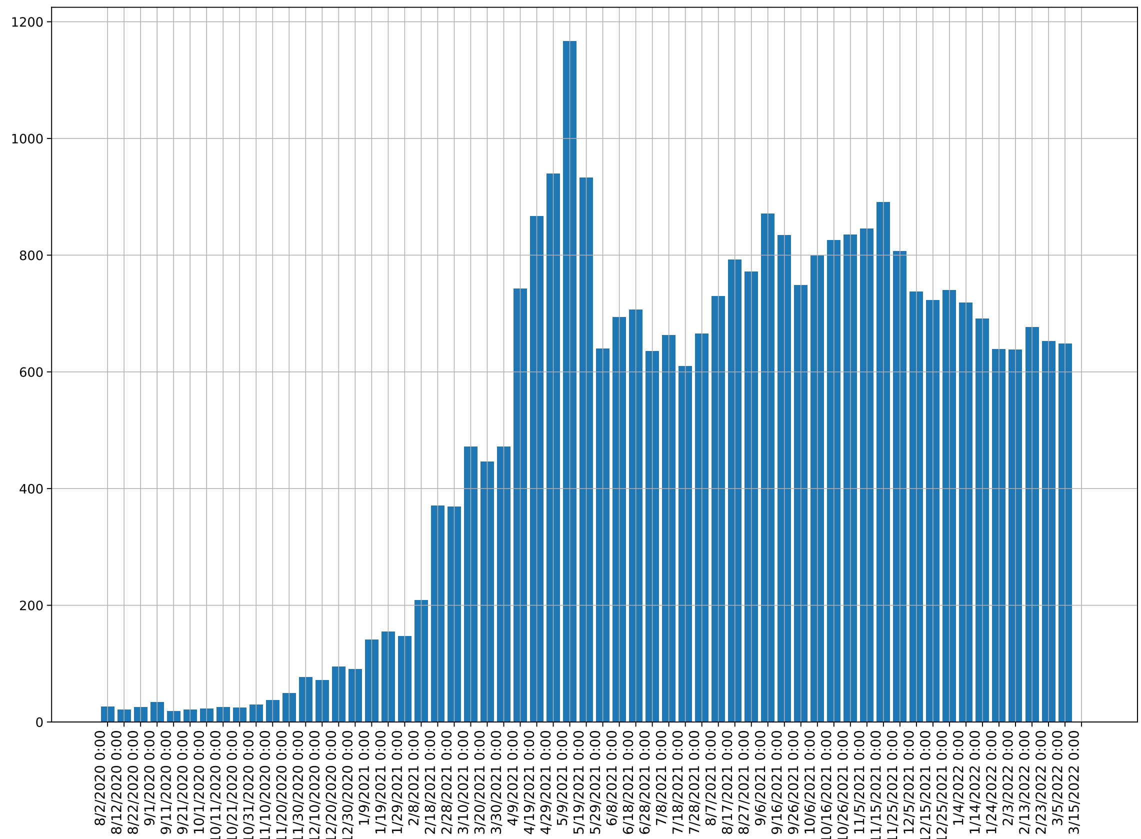


Figure 19: Equal spread result with 6 coins.

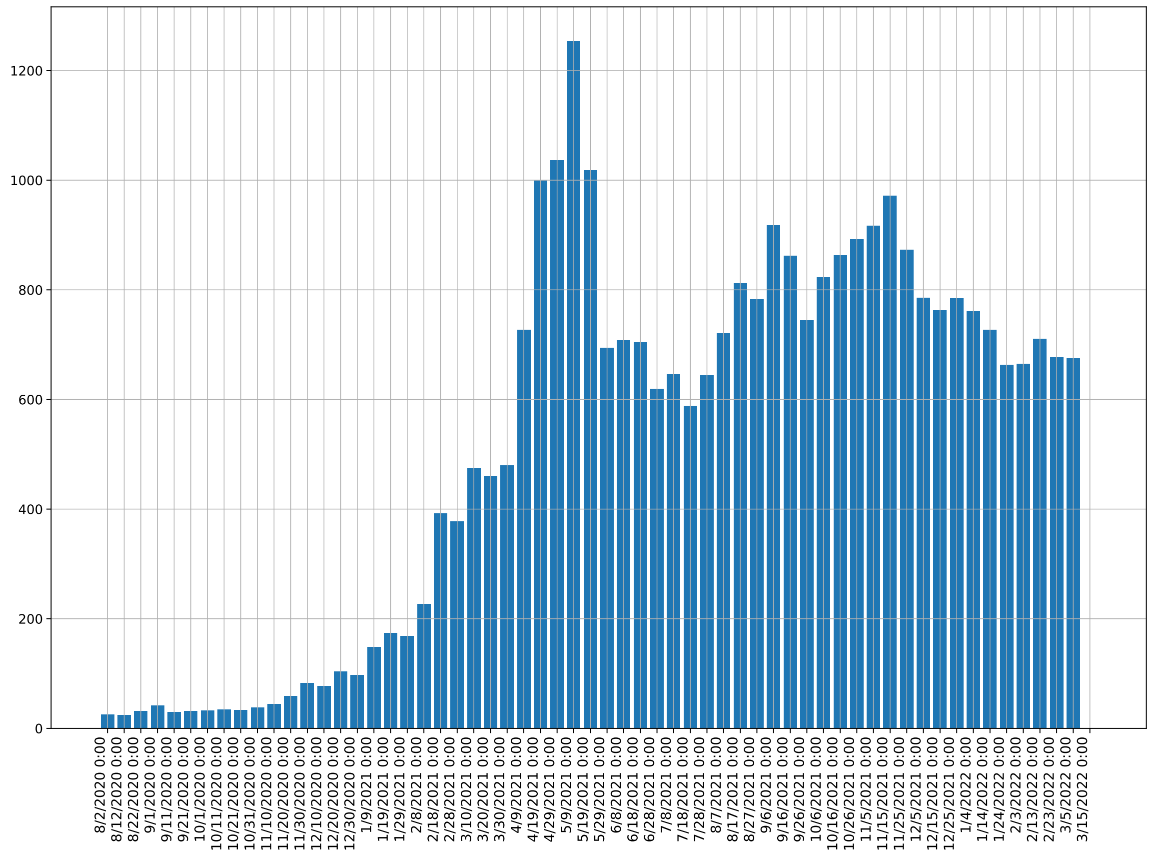


Figure 20: Equal spread result with 7 coins.

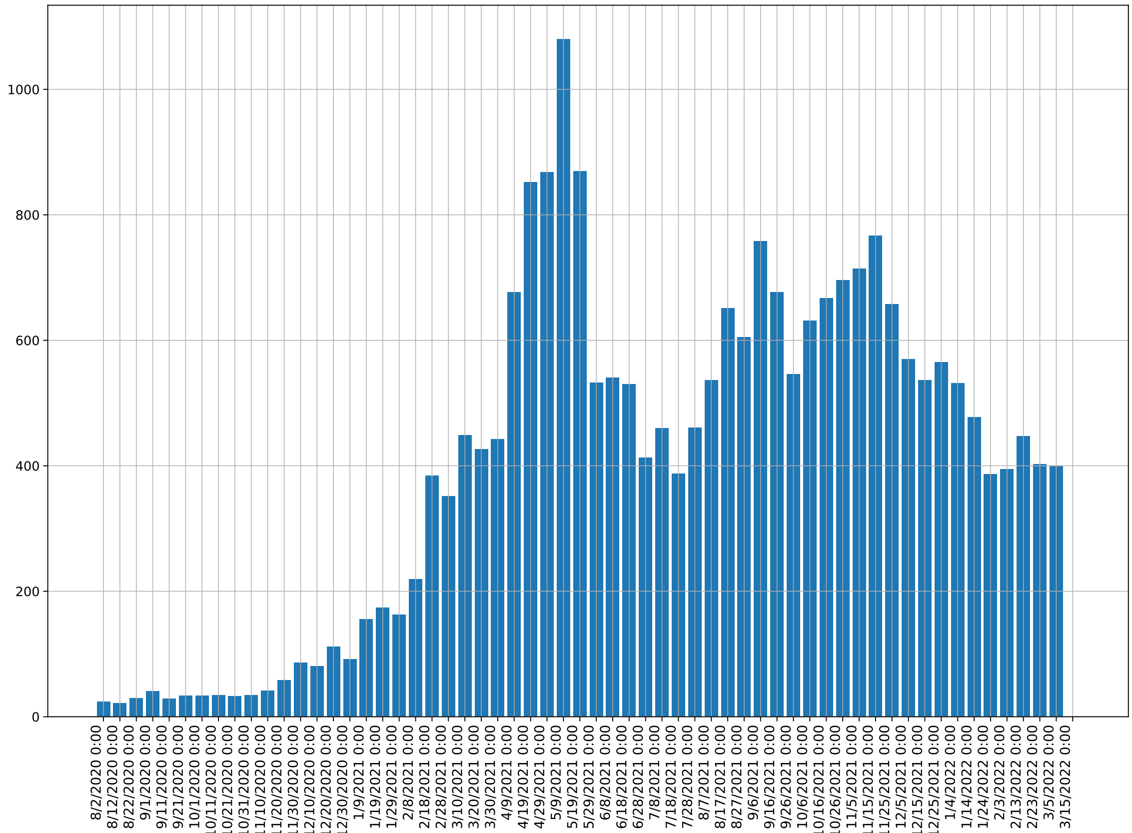


Figure 21: Equal spread result with 8 coins.

6 Discussion

The goal of this research was to compare most researched prediction methods and test their effectiveness in the real-world scenario, and test possibility of creating the profitable software package for crypto trading.The results are that RNNs can be quite useful, and they can give satisfying results for the users. RNNs also give all similar results, in some cases same results, this shows that they are almost equal when predicting the momentum of growth/fall of the value of coin, which is the most important aspect when predicting the crypto currencies.

Although these are not risk-free methods regarding profit. As we can see in the results the profitability can vary greatly. Periods of time where coins are growing will of course bring more profit, but the prediction methods can recognize the dips and the trading strategies/algorithms stop trading to avoid loss. Also one notable detail is that prediction methods can predict the momentum wrong, so we get trades which lose money, so there is no fail-proof method/strategy which works in every single scenario. As we can see in results, some TS-coin-PM combination can bring catastrophic losses, figures in Appendix A show us that it is possible to lose +200% of profit. From that we have learned that with unlucky timing, we can destroy portfolio, luckily, even that combination has been able to recover. This means that coin markets can be really unpredictable even for RNNs. Overall, even with the unpredictability, the results are promising, since there wasn't a single method/strategy/coin combination which didn't yield positive profit over 600 days of simulation.

What must be mentioned is that some coins have only risen in the value tremendously without having much of a drop off, an example being the BNB. Another thing to point out is that most of the coins have risen in value considerably during the beginning of the 2021, and all coins used in research have risen in value during the time period tested. This fact might have affected results to be more optimistic, but the counter argument against that is the fact that in WD strategy across all prediction methods, 7/9 coin have shown improvement over simply holding the coin. The results have also shown that TR strategy has achieved higher profit percentage than buy and hold average, same goes for the ES strategy. These are promising results for future work and improvement of this project.

7 Conclusion and future work

We had 3 parts of our research. First part was predicting the models in which we have found out that, all 3 RNN models are quite similar and they yielded similar results. BiLSTM is the most accurate model out of these three and it had best MAPE. What we have learned also is that most important aspect when predicting the crypto values is the momentum recognition, as all 3 methods, while having differences in MAPE, achieved similar or, in some cases, the exact same results. This has shown us that the MAPE is not the sole indicator of the success of the prediction method. For future work regarding this we can focus on further improving the RNN methods used and figure out the effects this has on simulation.

For trading strategies part of the research, the ES strategy gave most profit at 3 coins owned at same time, however, this was hugely affected by BNB's high growth, which is an outlier in this research and we can see results with BNB excluded from simulation in Appendix B, which are more realistic. Regarding TR strategy, we have concluded that it is inferior to the ES strategy because ES is safer (the money is spread out to more coins) and makes creates bigger profit. Regarding WD strategy, we concluded that it can be safe and profitable, because it rarely lost money, so creating hybrid between ES and WD might be something that can yield even better results.

In the simulation part of the research we have found out that there are trades that lose money (sometimes massive amounts), so the key finding here is that there is no risk-free strategy/prediction method pair that is completely superior, the reason for this being the unpredictability of the coin markets, in which there is no single coin that can be predicted with a high amount of confidence.

That has led us to the another finding, that coins can vary in predictability. We have learned this by looking at the both MAE results and the simulation results where we see profit percentages. Explanation for this is simply that different coins have the different following. This can be a theme for future research what types of coin are easier/harder to predict, since they all varied quite a bit, also another theme for future research is how to set up models for certain coins, and whether that makes any significant difference.

Another founding in this research is that it is possible to create monetary gain from using open-source software, although risky, it is still net positive. Thus, this research can be basis for creating more similar and improved software.

8 Povzetek naloge v slovenskem jeziku

Kriptovalute so močno spremenile svetovno finančno okolje. Postale so ena od tveganih naložb. Ker tržna kapitalizacija pri najbolj priljubljenih kovancih sega od 10 do 500 milijard ameriških dolarjev, so lahko kriptovalute glede na določene pogoje dobra naložba. Pomembno je opozoriti, da je trg zelo spremenljiv (volatilen), zaradi česar je zelo nestabilen. To pomeni, da sta padec ali skok njihovih cen nekoliko nepredvidljiva, kar lahko pomeni višje dobičke ob večjem tveganju.

Ustvarjanje dobička pri trgovanju s kriptovalutami je v primerjavi z običajnim trgovanjem z delnicami veliko težja naloga zaradi več dodatnih dejavnikov. Dejstvo je, da so trgi kriptovalut zelo nepredvidljivi in nanje lahko vpliva vsak svetovni dogodek. Dejavniki, ki prispevajo k nestanovitnosti kriptovalut so nepodprtost, zato so njihove vrednosti odvisne izključno od trgovcev, kar pomeni, da v nasprotju z delnicami nimajo notranje vrednosti. To pomeni, da je napovedovanje in pridobivanje dobička iz vrednosti kriptovalut morda težje kot pri delnicah.

V tej študiji želimo odpraviti vrzel, ki obstaja v večini študij v zvezi z napovedovanjem kriptovalut, in sicer pomanjkanje simulacijskega okolja, v katerem bi lahko resnično preizkusili metode napovedovanja in način, kako lahko uporabimo njihova priporočila. Natančneje, uporabili bomo priporočilni mehanizem v povezavi z nevronsko mrežo s povratno zanko (RNN), naučeno s predhodnimi podatki, ki vključujejo kriptovalute in njihove zaključne cene. Predlagani pristop nameravamo oceniti z izvajanjem simulacij kripto-poslovnih transakcij na podlagi preteklih podatkov, da bi našli optimalen način uporabe metod napovedovanja in njihovih priporočil.

V naši raziskavi smo imeli tri dele. Prvi del je bil napovedovanje modelov, pri katerem smo ugotovili, da so si vsi trije modeli RNN precej podobni in so dali podobne rezultate. BiLSTM je najnatančnejši model od teh treh in je imel najboljši MAPE od vseh treh metod. Ugotovili smo tudi, da je najpomembnejši vidik pri napovedovanju vrednosti kriptovalut prepoznavanje zagona, zato so vse tri metode kljub razlikam v MAE dosegle podoben ali v nekaterih primerih popolnoma enak rezultat, kar nam je pokazalo, da MAE ni edini pokazatelj uspešnosti metode napovedovanja. Pri prihodnjem delu v zvezi s tem se lahko osredotočimo na nadaljnje izboljšanje uporabljenih metod RNN in ugotovimo, kakšen vpliv ima to na simulacijo.

Pri delu raziskave, ki se nanaša na strategije trgovanja, ima najpreprostejša strategija (Waiting out the dip) na splošno najboljše rezultate, čeprav je morda najbolj tvegana. V tabeli z rezultati 3 lahko vidimo, da je strategija WD uspešna, drugi dve strategiji pa sta prinesli več dobička kot držanje, če povprečimo dobičke držanja za 9 kovancev. To pomeni, da je katera koli od treh strategij uspešna, čeprav je strategija ES boljša od strategije TR, zato je izbira med strategijama WD in ES odvisna od osebnih preferenc. Ta del je mogoče izboljšati z oblikovanjem hibridne strategije z WD in ES, ki bi lahko dala boljše rezultate.

V simulacijskem delu raziskave smo ugotovili, da obstajajo posli, ki izgubijo denar, zato je ključna ugotovitev, da ni para strategije/metode napovedovanja brez tveganja. Razlog za to je nepredvidljivost trgov kovancev, na katerih ni nobenega kovanca, ki bi ga bilo mogoče napovedati z veliko mero zaupanja.

To nas je pripeljalo do druge ugotovitve, da je različne kovance težje ali lažje napovedati. To smo ugotovili z ogledom rezultatov MAE in rezultatov simulacije, kjer so prikazani odstotki dobička. Razlaga za to je preprosto ta, da imajo različni kovanci različno sledenje. To je lahko tema za prihodnje raziskave, katere vrste kovancev je lažje/težje napovedati, saj so se vsi precej razlikovali, prav tako je druga tema za prihodnje raziskave, kako nastaviti modele za določene kovance in ali to pomeni kakšno pomembno razliko.

Zadnja ugotovitev te raziskave je, da je mogoče z uporabo odprtokodne programske opreme ustvariti denarni dobiček, čeprav je tvegan, je še vedno neto pozitiven. Tako je ta raziskava lahko podlaga za ustvarjanje izboljšane programske opreme.

9 Bibliography and Sources

[1] Khalid Abouloula and Salah-ddine Krit. Using a Robot Trader for Automatic Trading. *ICEMIS'18, June 19–21, 2018, Istanbul, Turkey*, 4:1–9, 2018. *(Cited on page 3.)*

[2] Jaysing Bhosale and Sushil Mavale. Volatility of select Crypto-currencies: A comparison of Bitcoin, Ethereum and Litecoin. *ISSN 2348–0661 Print © 2018) Symbiosis Centre for Management Studies, Pune Annual Research Journal of Symbiosis Centre for Management Studies, Pune Vol. 6, March 2018*, 6:10, 2018. *(Cited on page 2.)*

[3] Florian FOLLONIER. Time Series Forecasting - Univariate Model using Recurrent Neural Networks, https://github.com/flo7up/relataly-public-python-tutorials/blob/master/003May 2022. *(Cited on page 9.)*

[4] Yiqing Hua. Bitcoin price prediction using ARIMA and LSTM. *E3S Web of Conferences*, 218:01050, 2020. *(Cited on page 3.)*

[5] Thomas E. Koker and Dimitrios Koutmos. Cryptocurrency Trading Using Machine Learning. *Journal of Risk and Financial Management*, 13(8):178, August 2020. *(Cited on page 4.)*

[6] Abhishek Kumar. Short-Term Prediction of Crypto-Currencies Using Machine Learning. *SSRN Electronic Journal*, 2:20, 2021. *(Cited on page 3.)*

[7] Yukun Liu and Aleh Tsyvinski. Risks and Returns of Cryptocurrency. *The Review of Financial Studies*, 6(6):2689–2727, May 2021. *(Cited on page 2.)*

[8] Sule Ozturk Birim. An analysis for cryptocurrency price prediction using lstm, gru, and the bi- directional implication. *Celal Bayar University Journal of Science*, 1:22, March 2022. *(Cited on pages 3, 7, and 8.)*

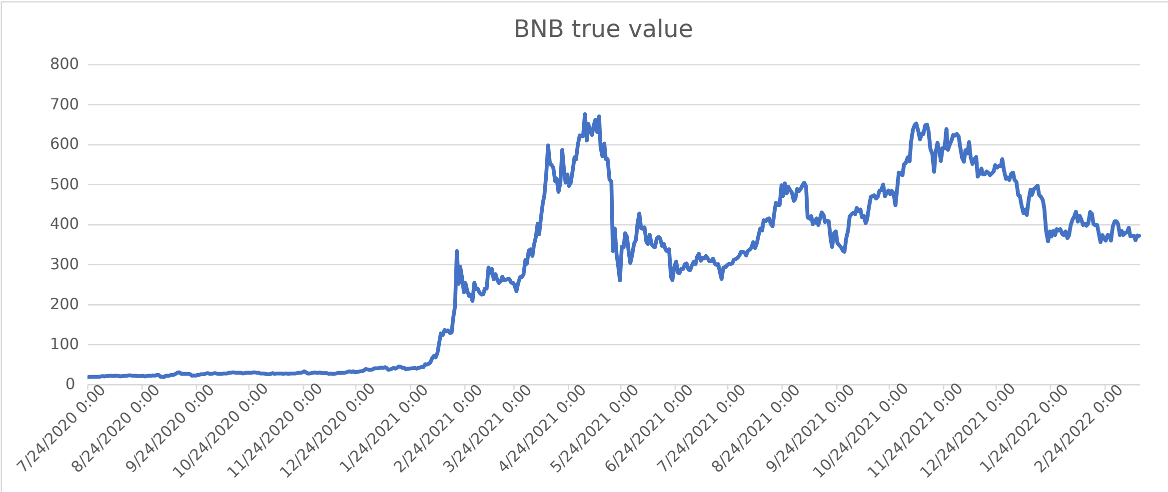
[9] Murtaza Roondiwala, Harshal Patel, and Shraddha Varma. Predicting Stock Prices Using LSTM. *International Journal of Science and Research (IJSR) ISSN (Online)*, 6(4):4, 2015. *(Cited on page 2.)*

[10] Peter T. Yamak, Li Yujian, and Pius K. Gadosey. A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting. *ACAI 2019: 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, 3:49–55, December 2019. *(Cited on page 3.)*

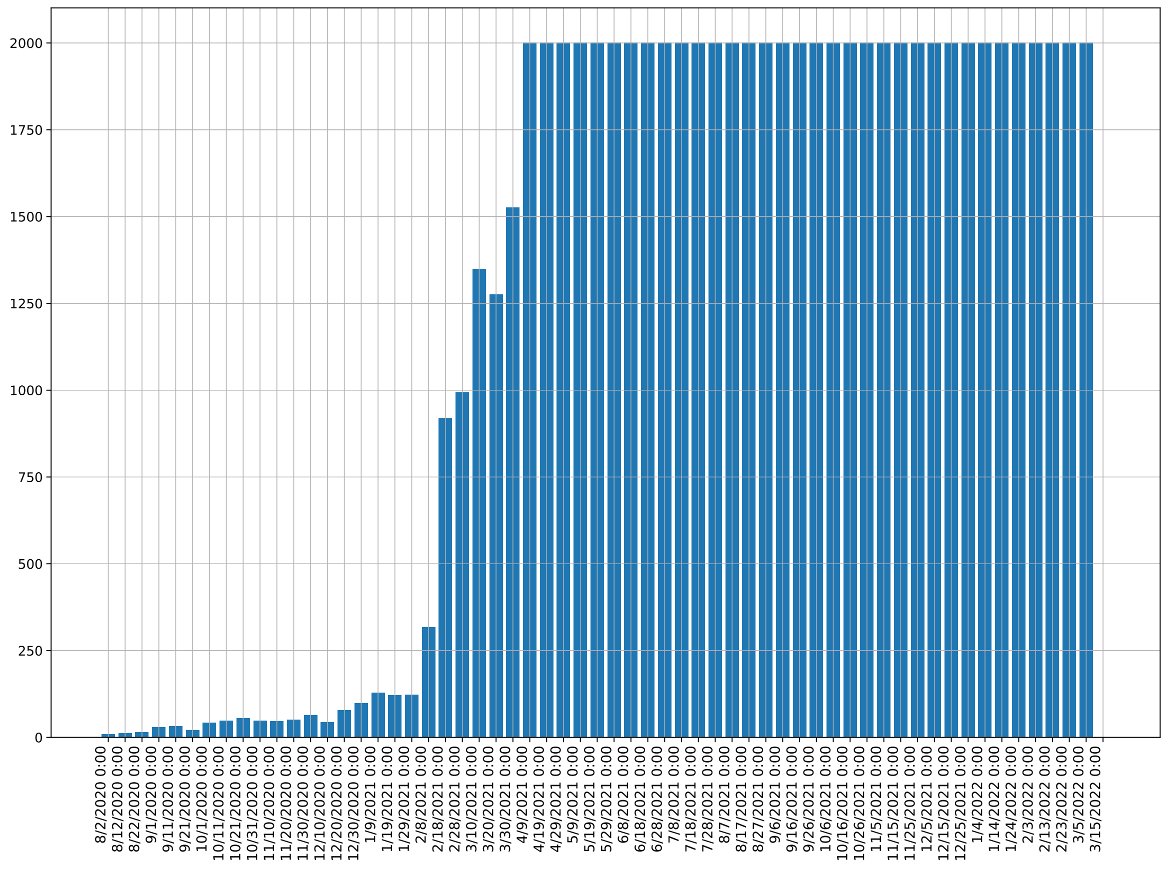
Appendices

A WD strategy all results

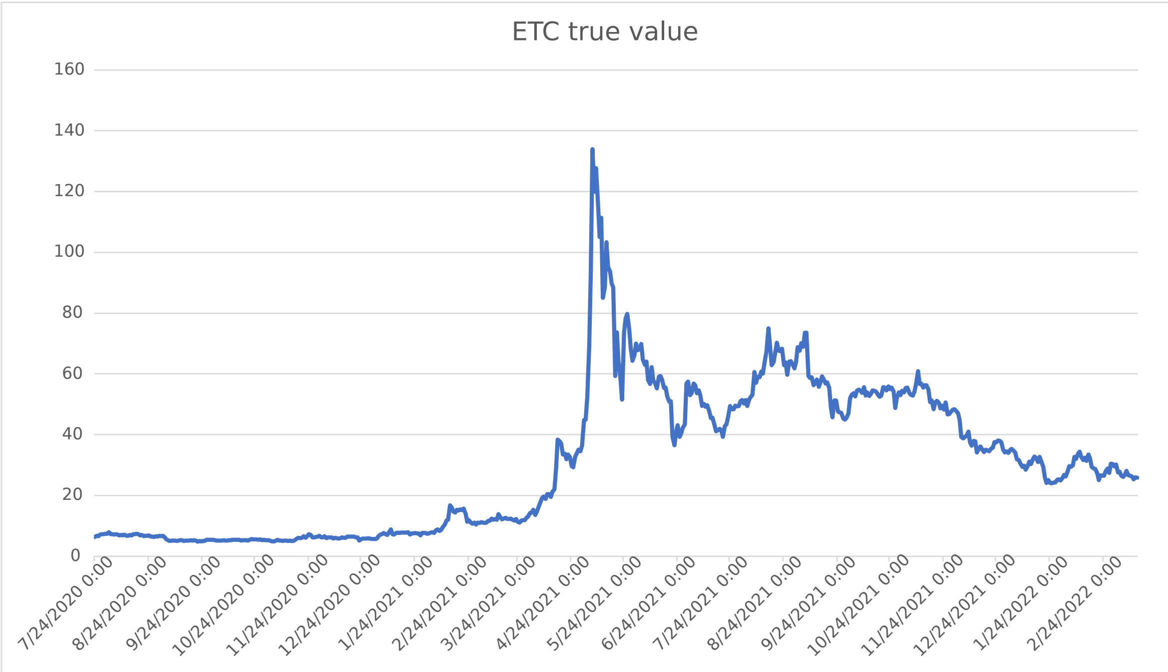
In this appendix we present results of WD strategy across all coins.



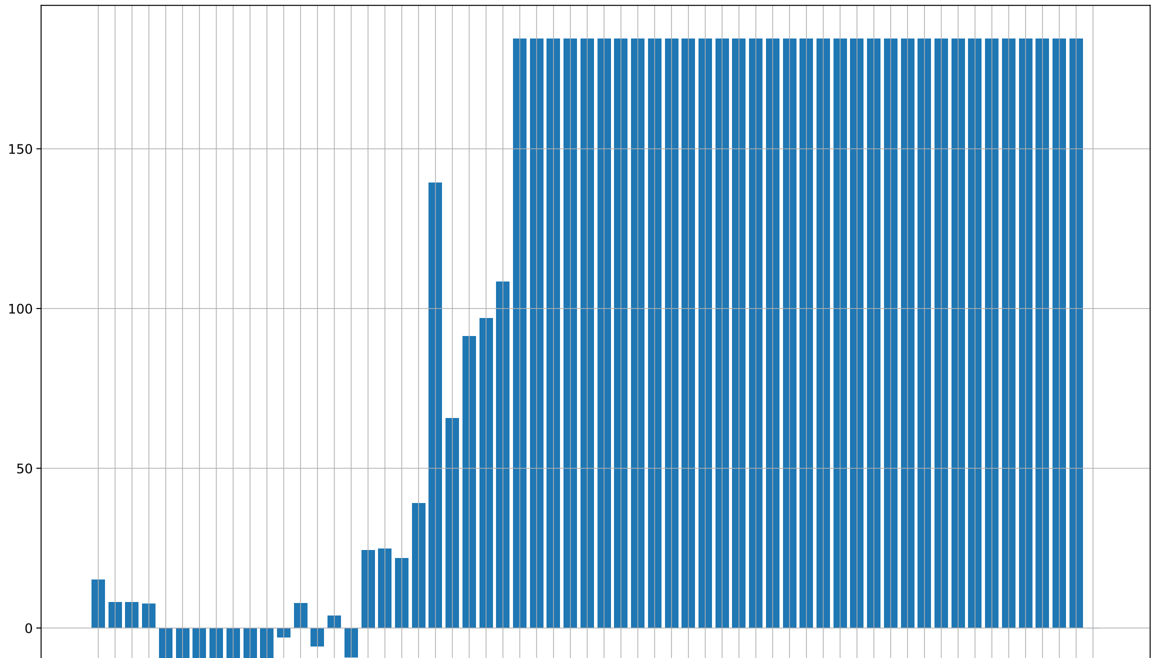
BNB Value over last 600 days.



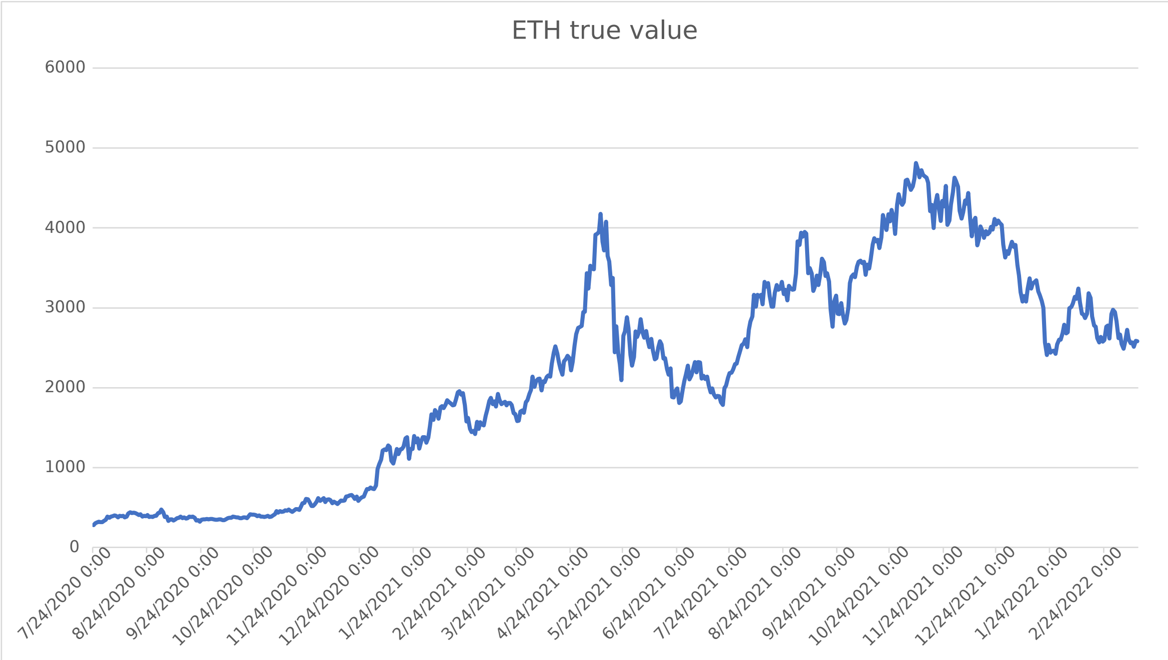
BNB Results with all 3 algorithms.



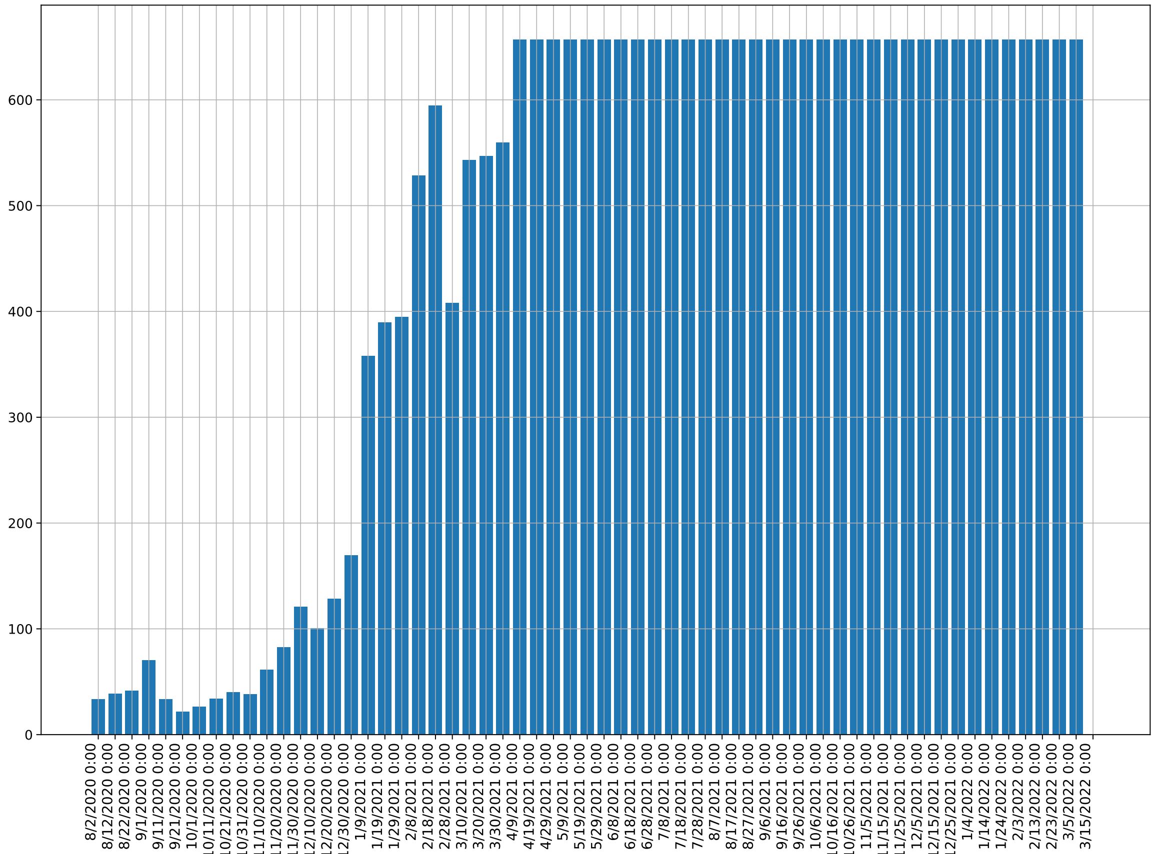
ETC Value over last 600 days.



ETC Results with all 3 prediction methods.

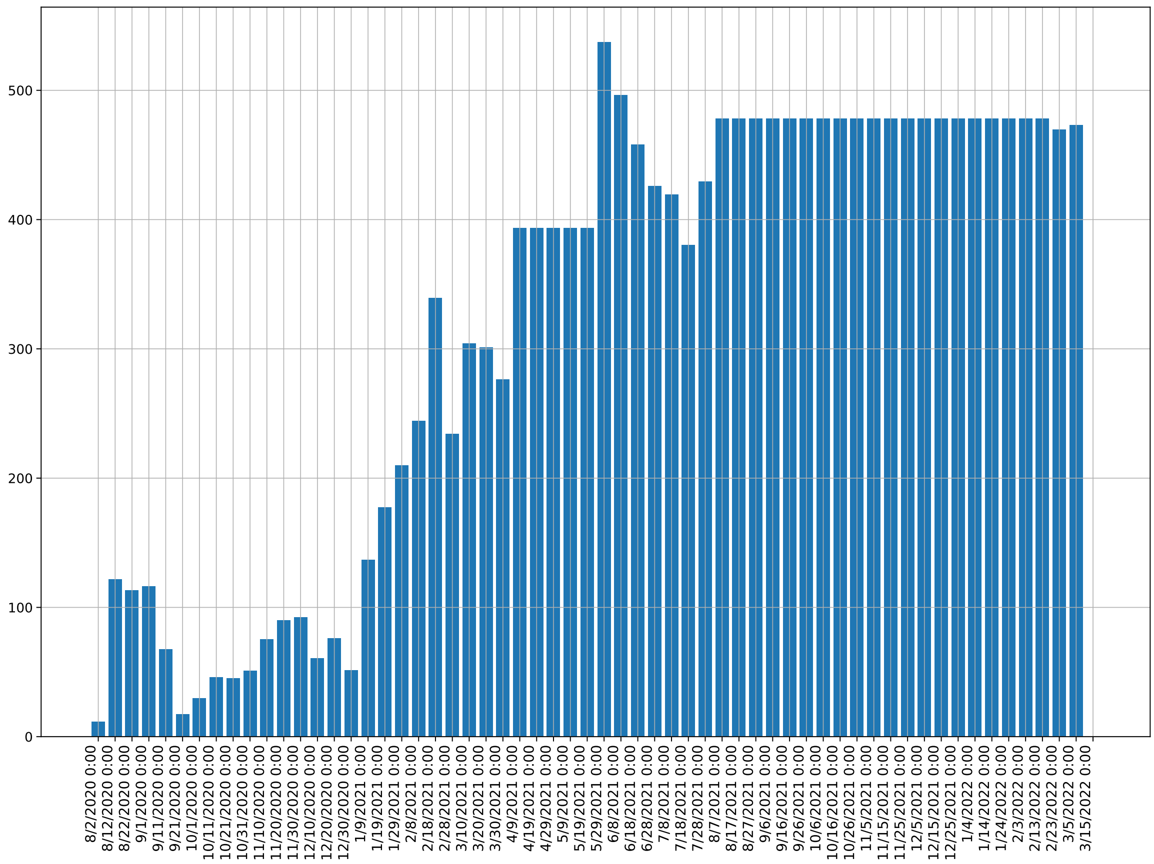


ETH Value over last 600 days.

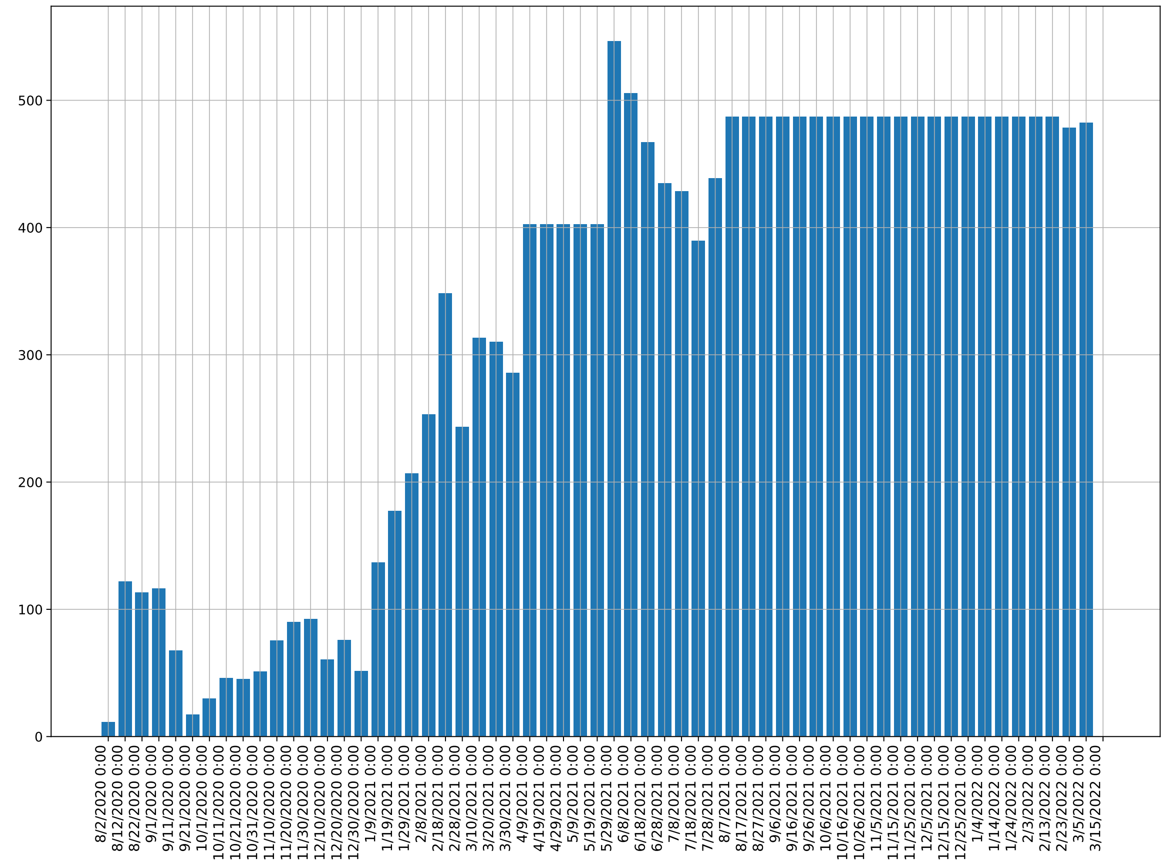


ETH Results with all 3 prediction methods.

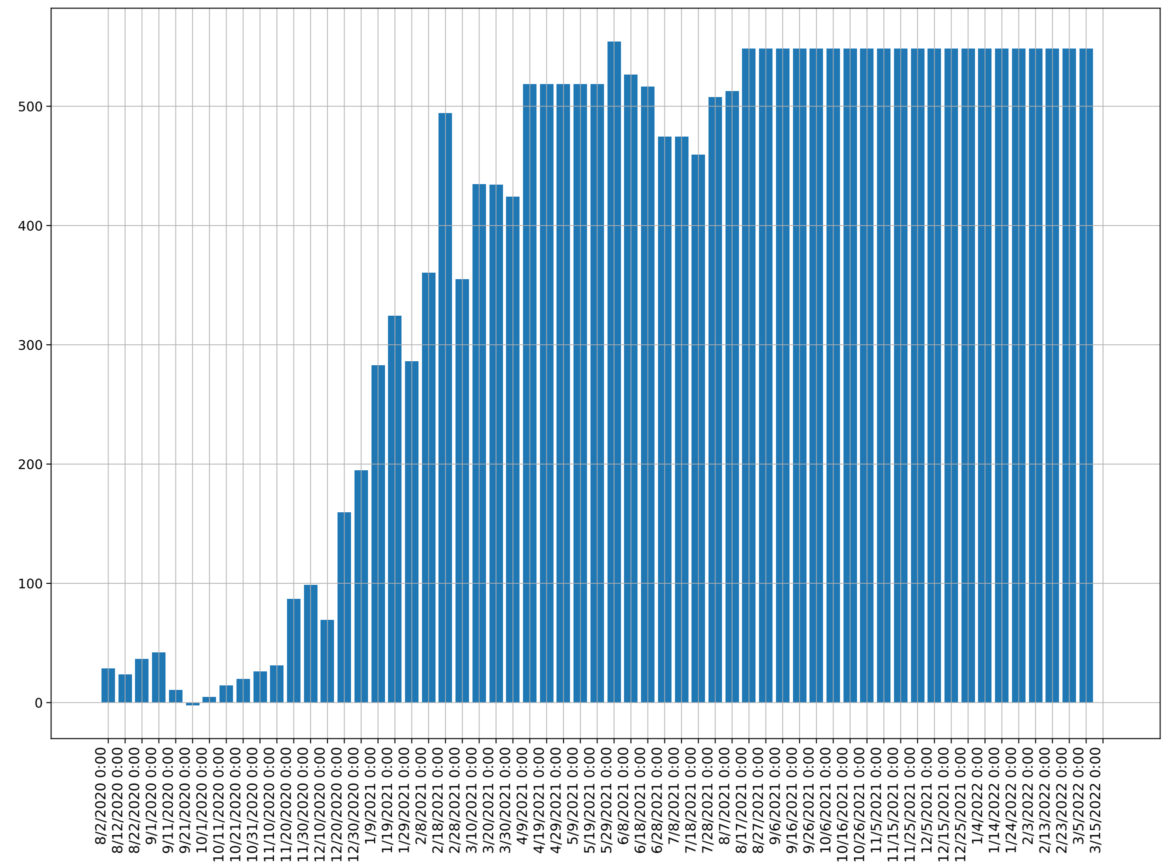
LINK Value over last 600 days.



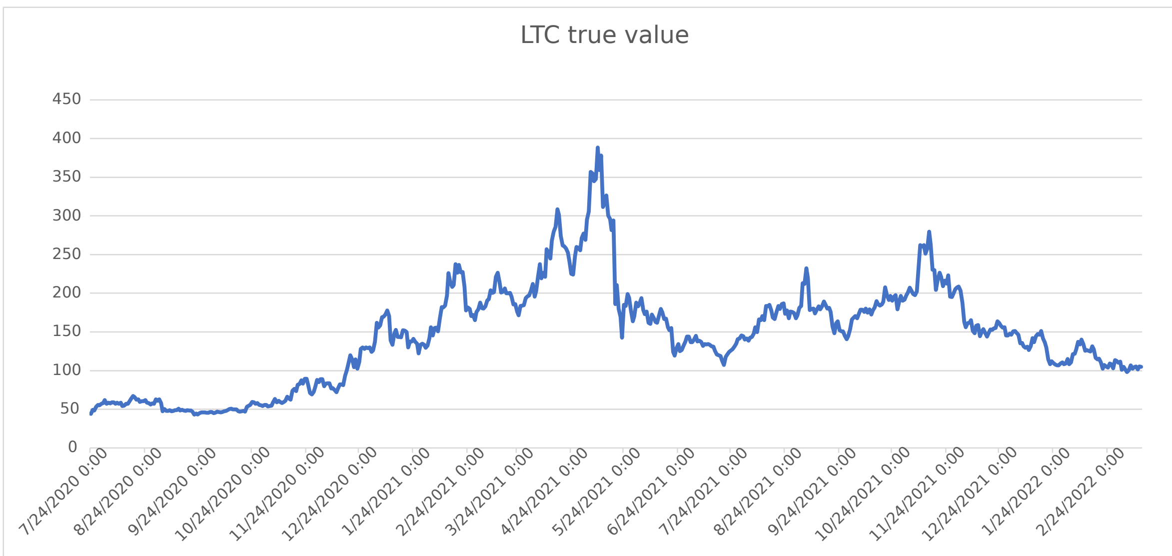
LINK results in profit percentage while using LSTM prediction method.



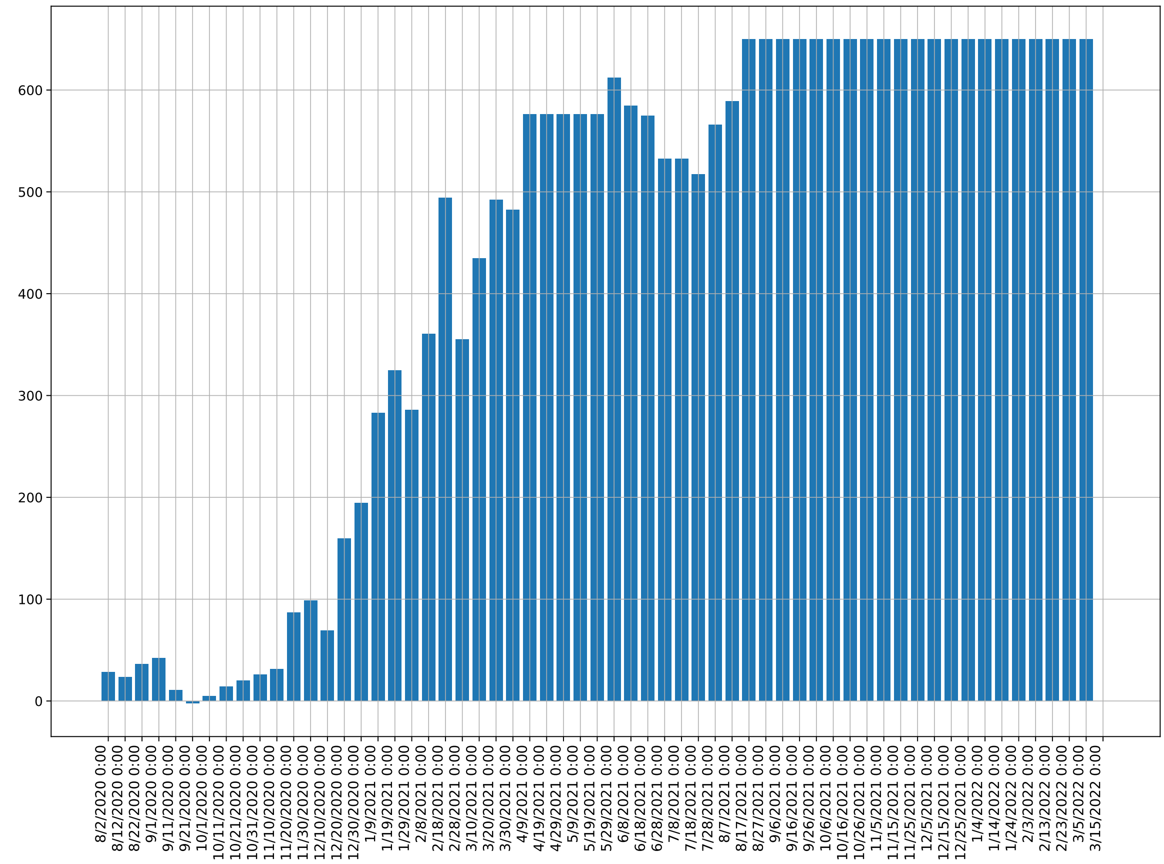
LINK results in profit percentage while using BILSTM prediction method.



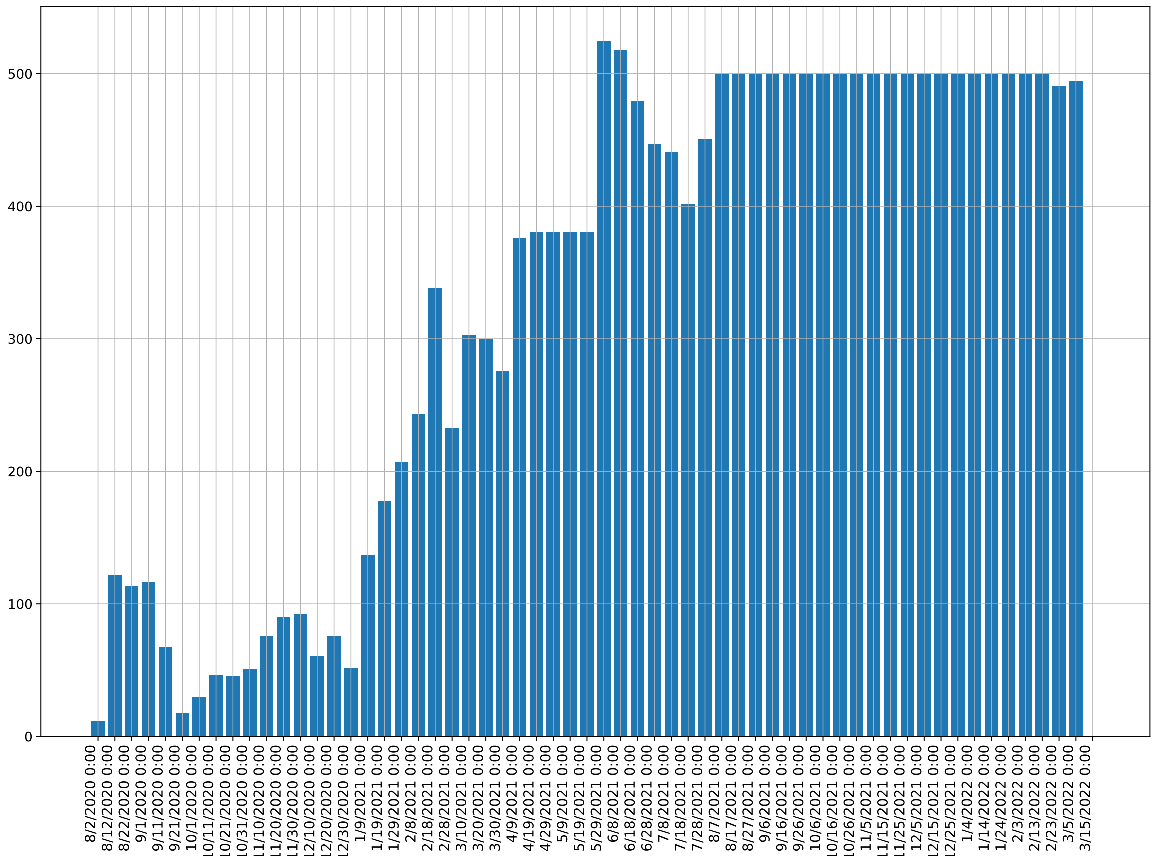
LINK results in profit percentage while using GRU prediction method.



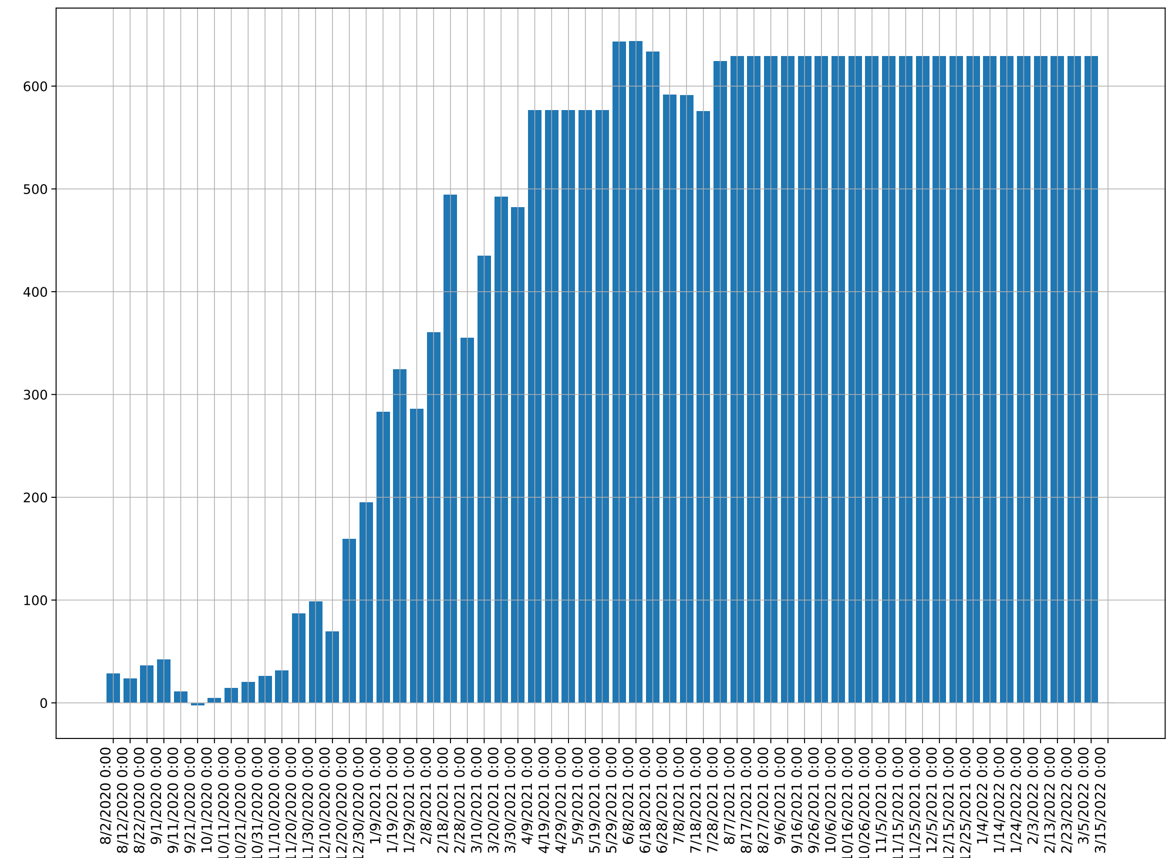
LTC Value over last 600 days.



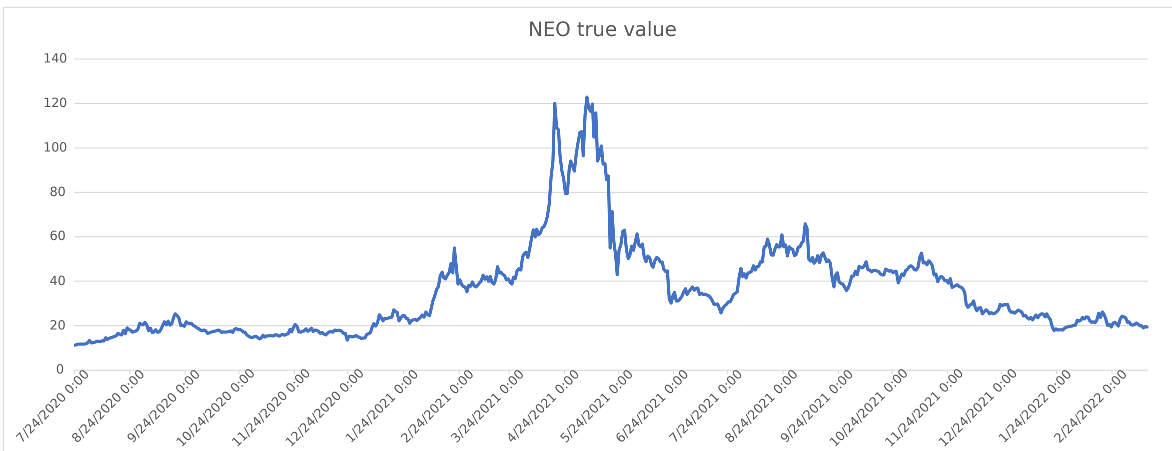
LTC results in profit percentage while using LSTM prediction method.



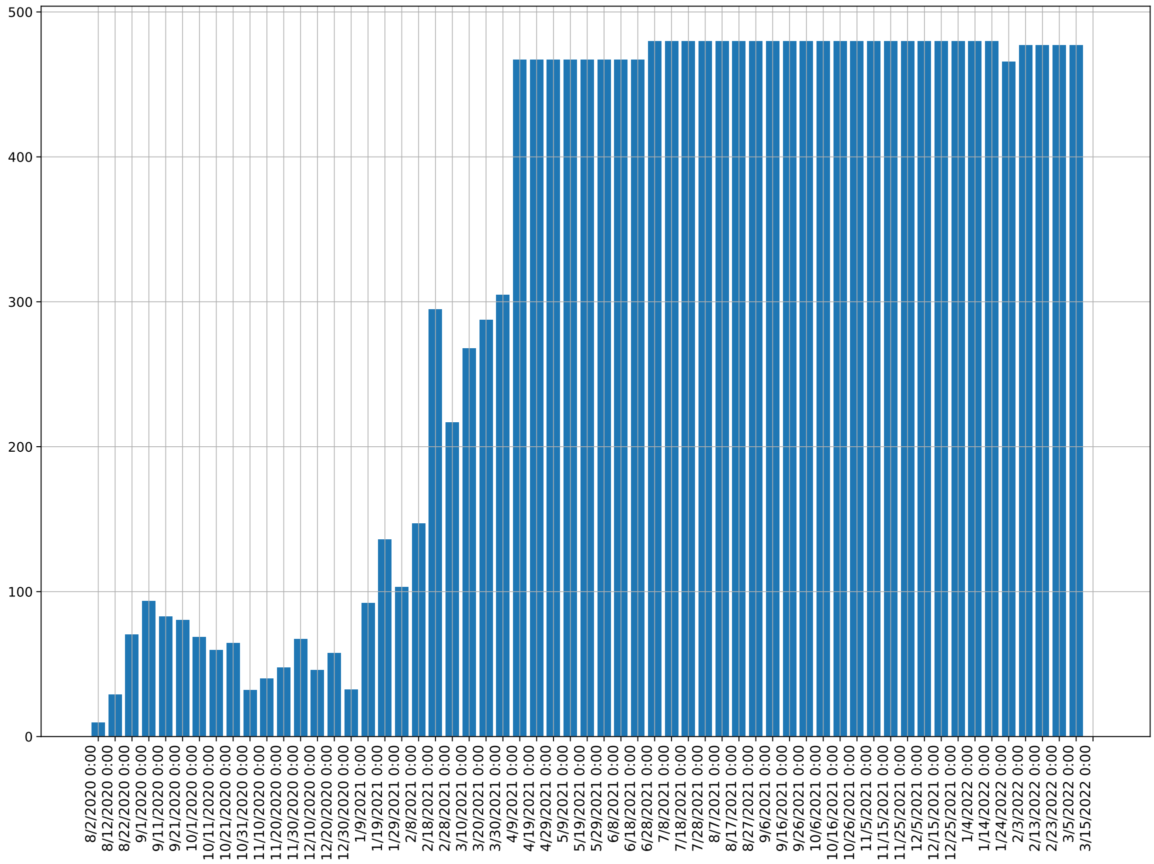
LTC results in profit percentage while using BILSTM prediction method.



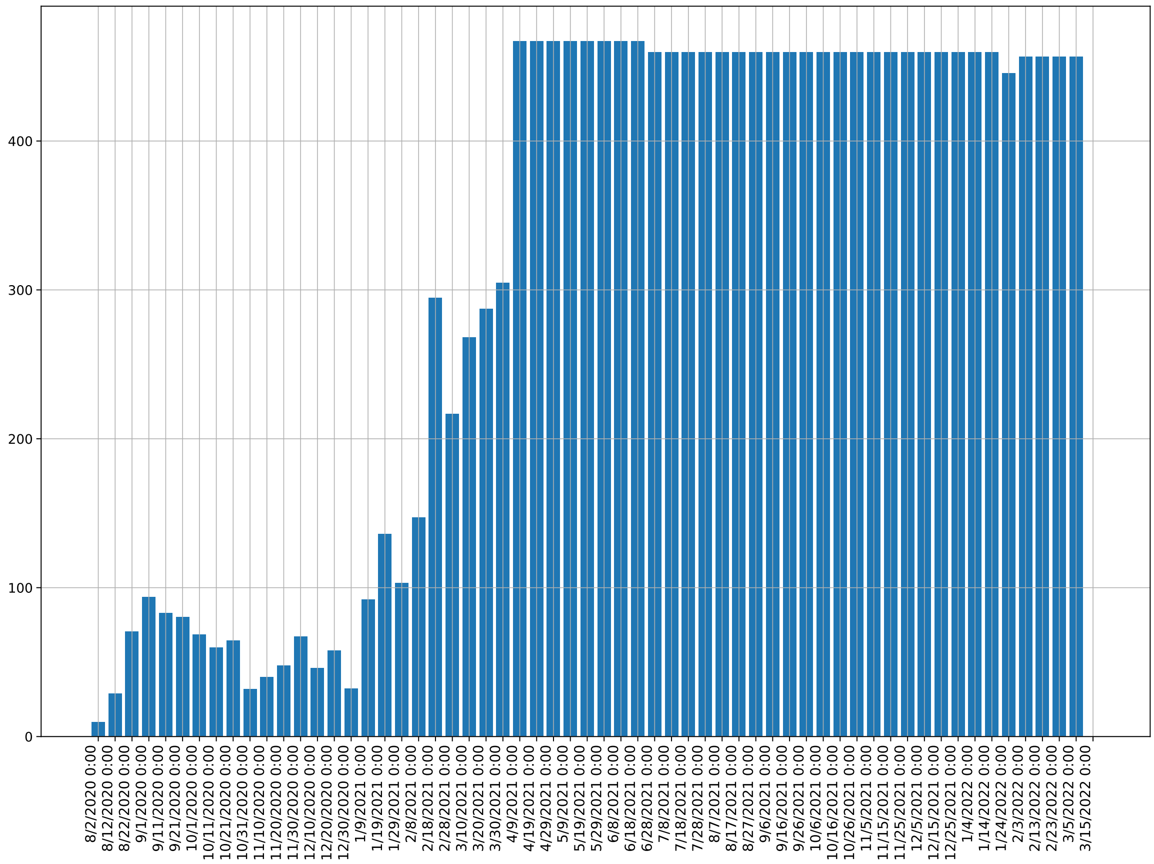
LTC results in profit percentage while using GRU prediction method.



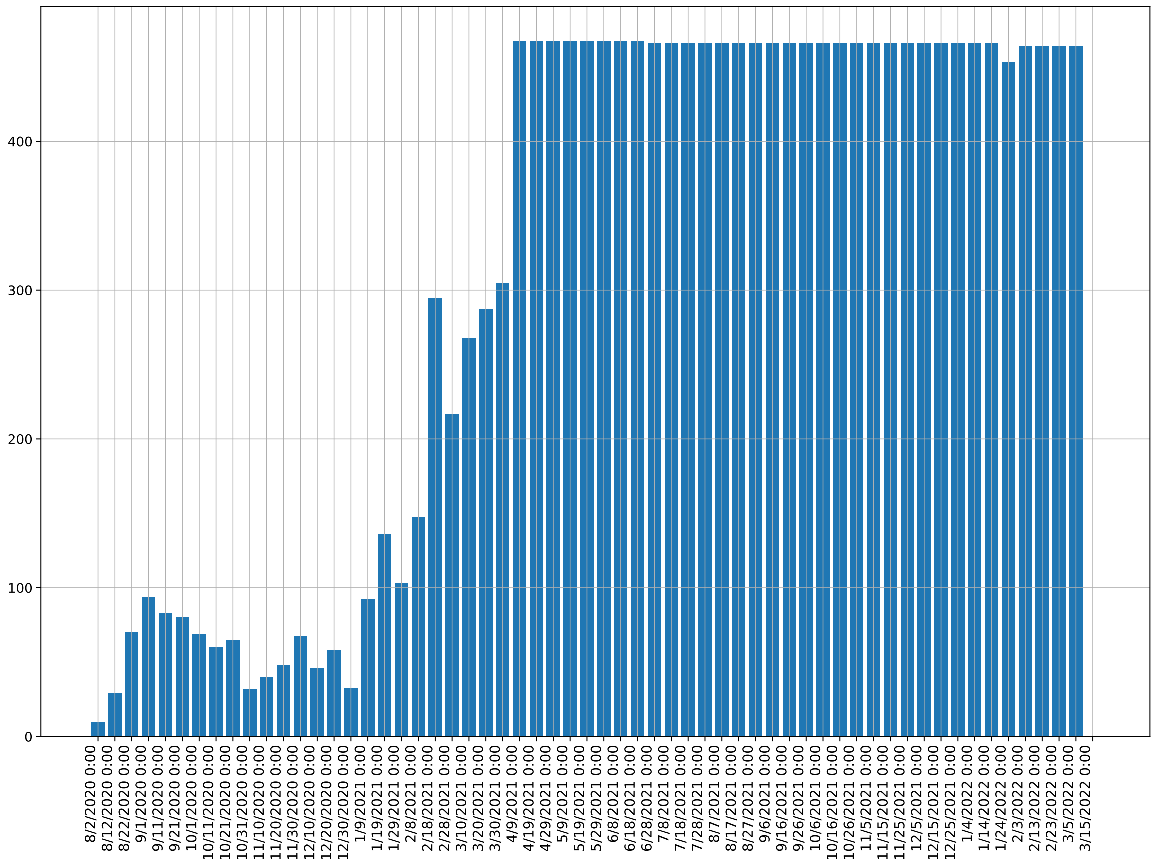
NEO Value over last 600 days.



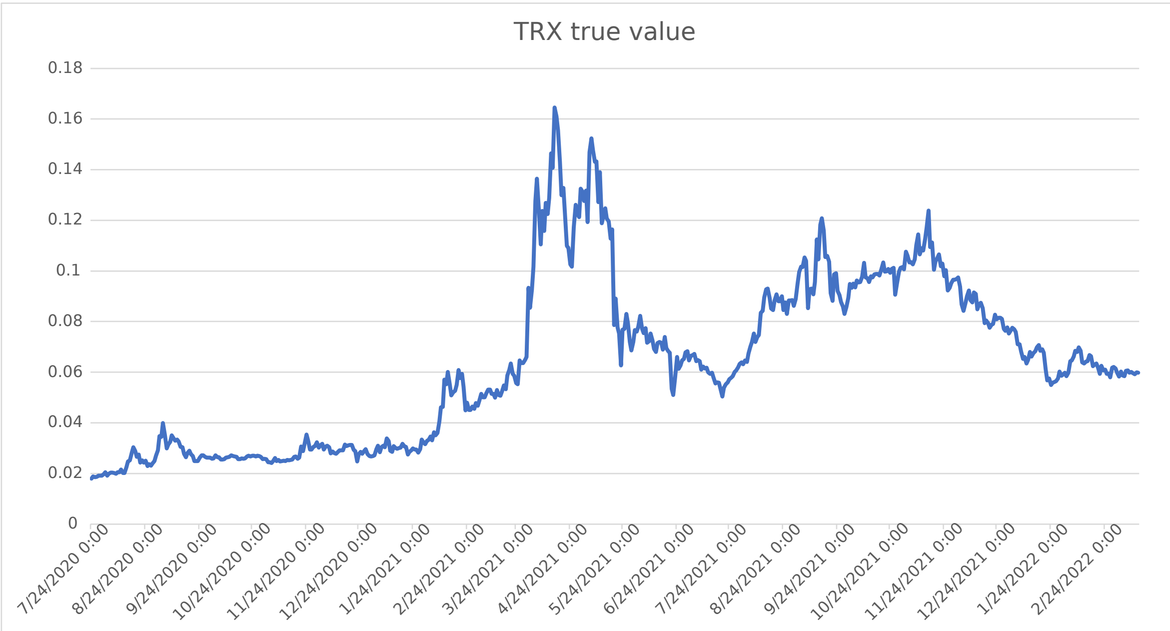
NEO results in profit percentage while using LSTM prediction method.



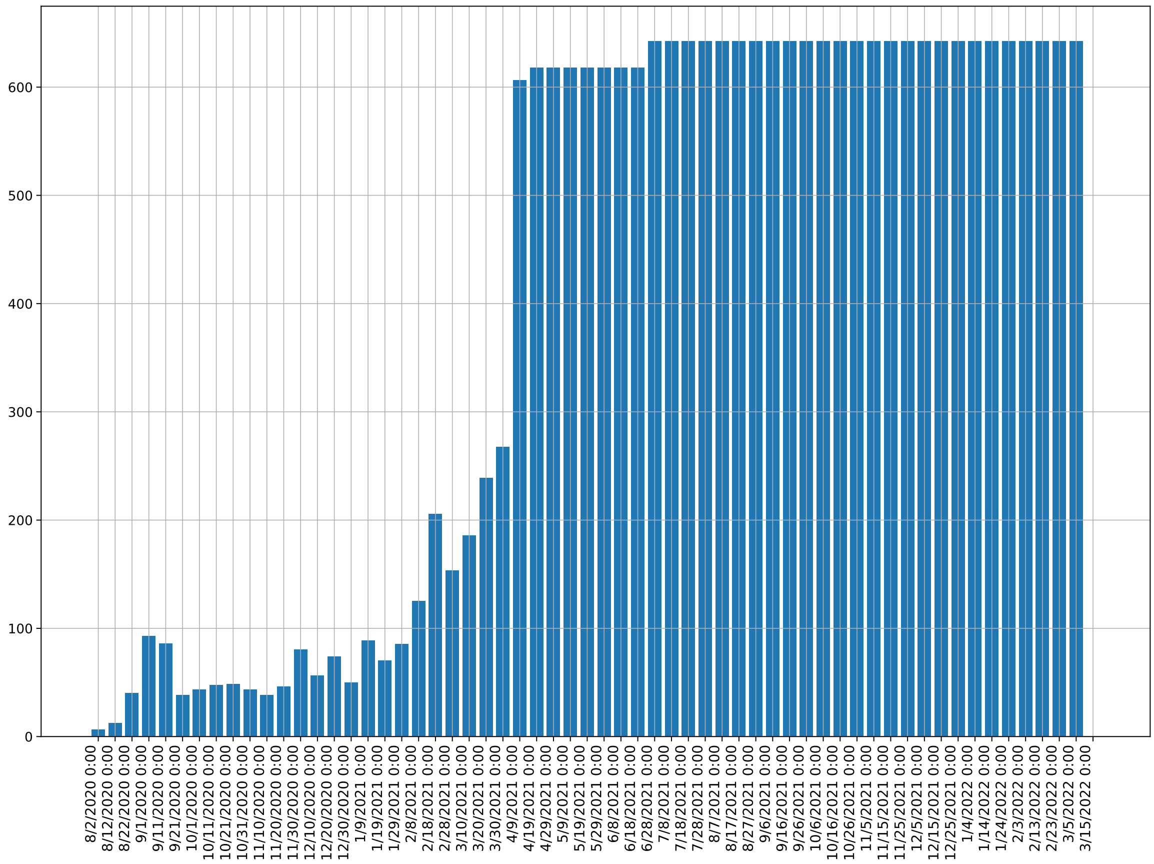
NEO results in profit percentage while using BILSTM prediction method.



NEO results in profit percentage while using GRU prediction method.

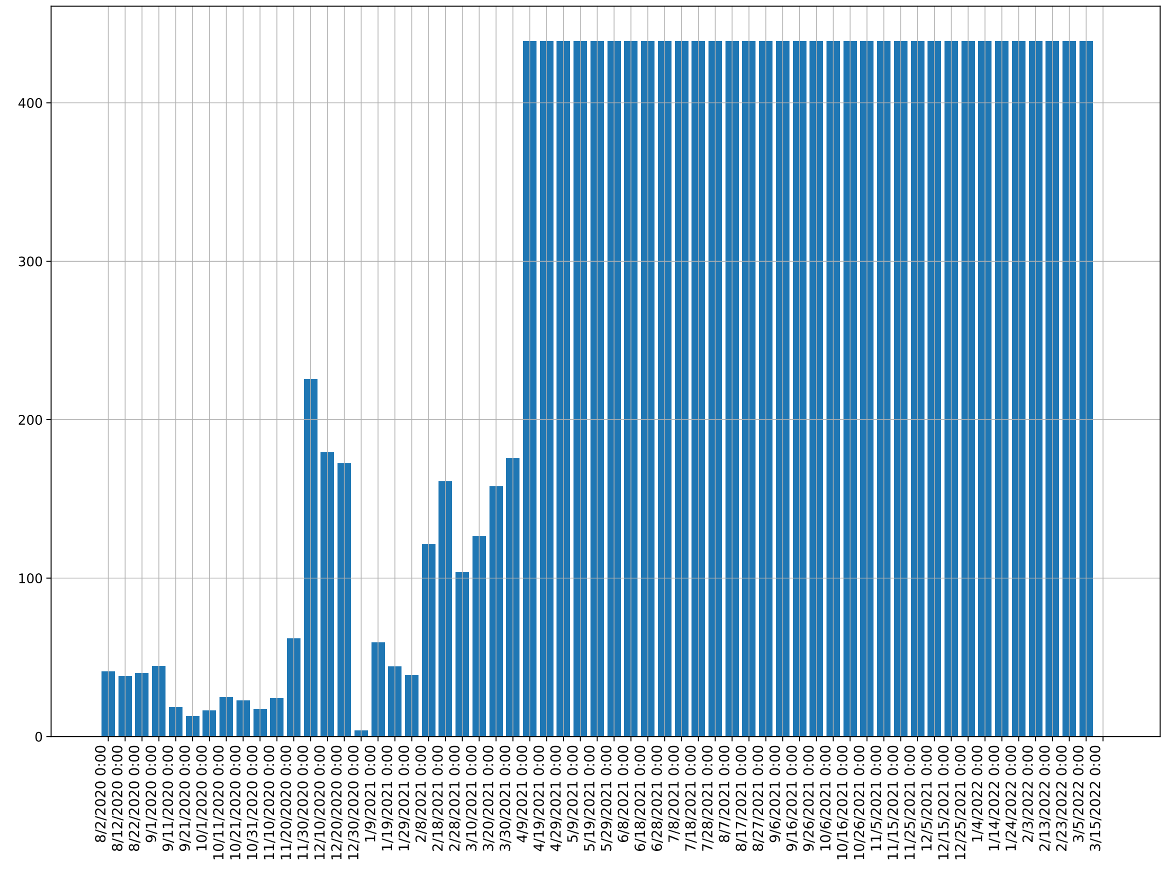


TRX Value over last 600 days.

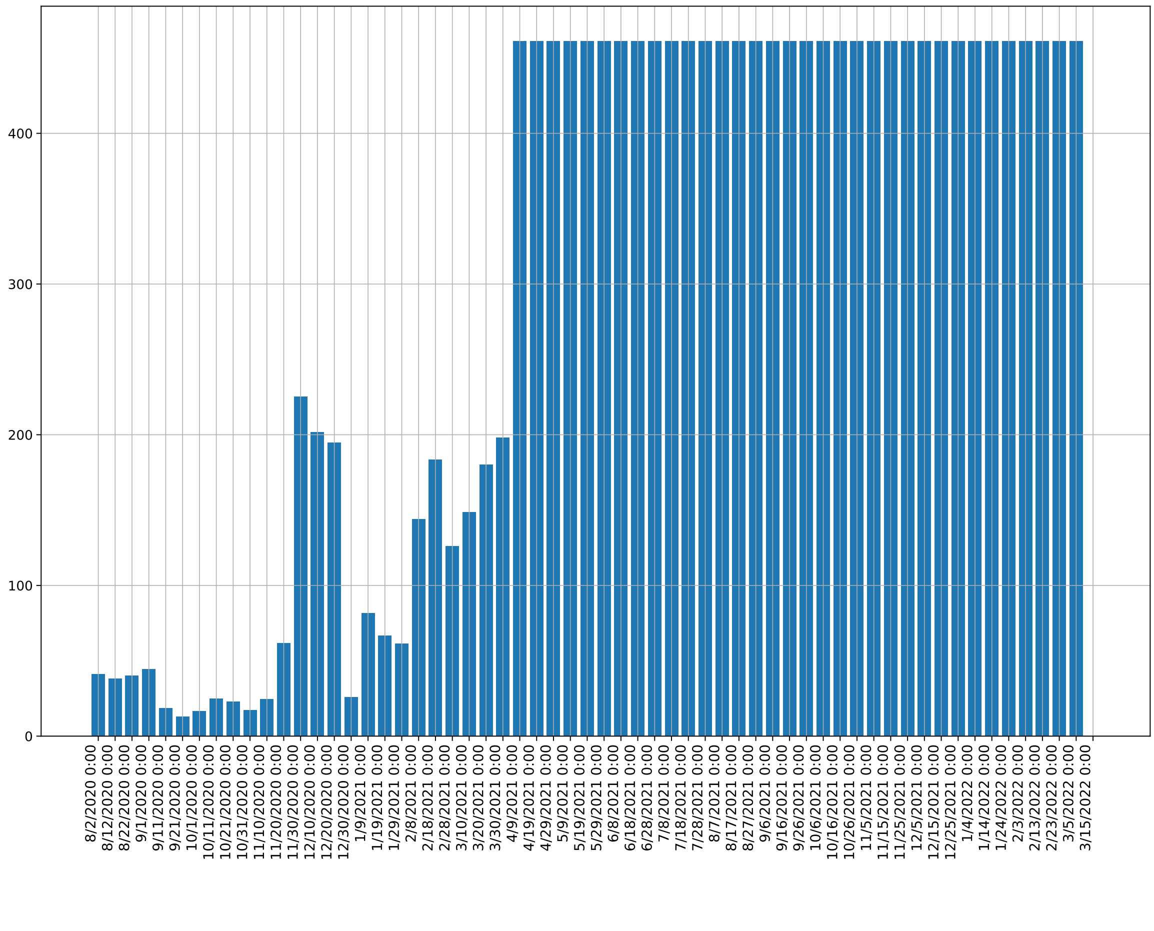


TRX Results with all 3 prediction methods.

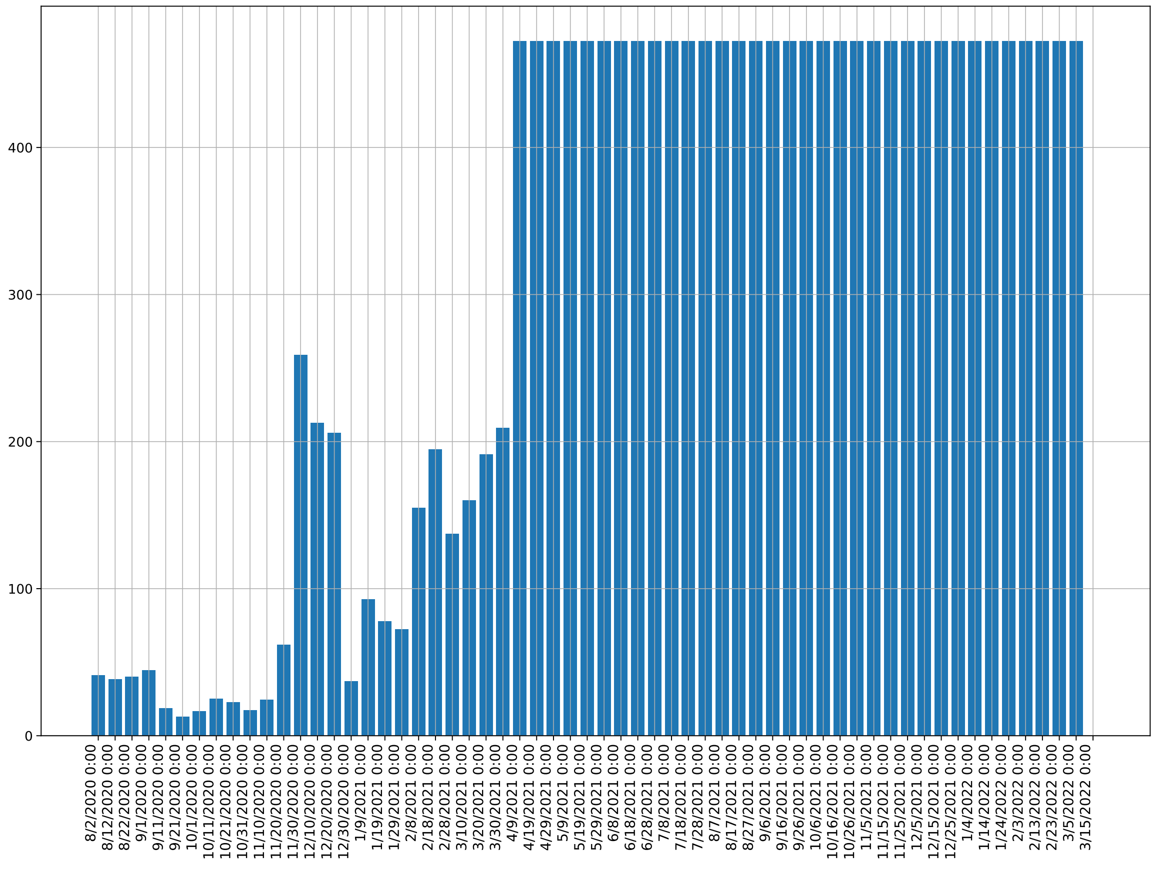
XRP Value over last 600 days.



XRP results in profit percentage while using LSTM prediction method.



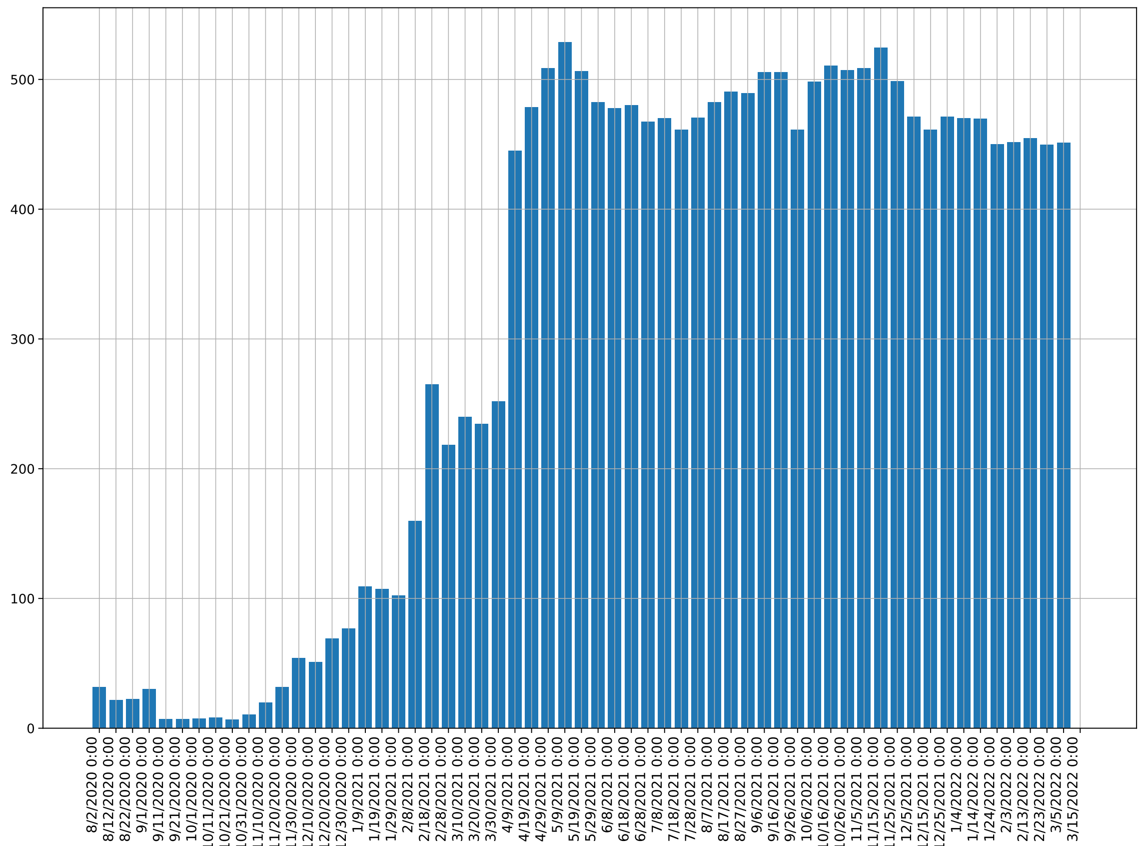
XRP results in profit percentage while using BILSTM prediction method.



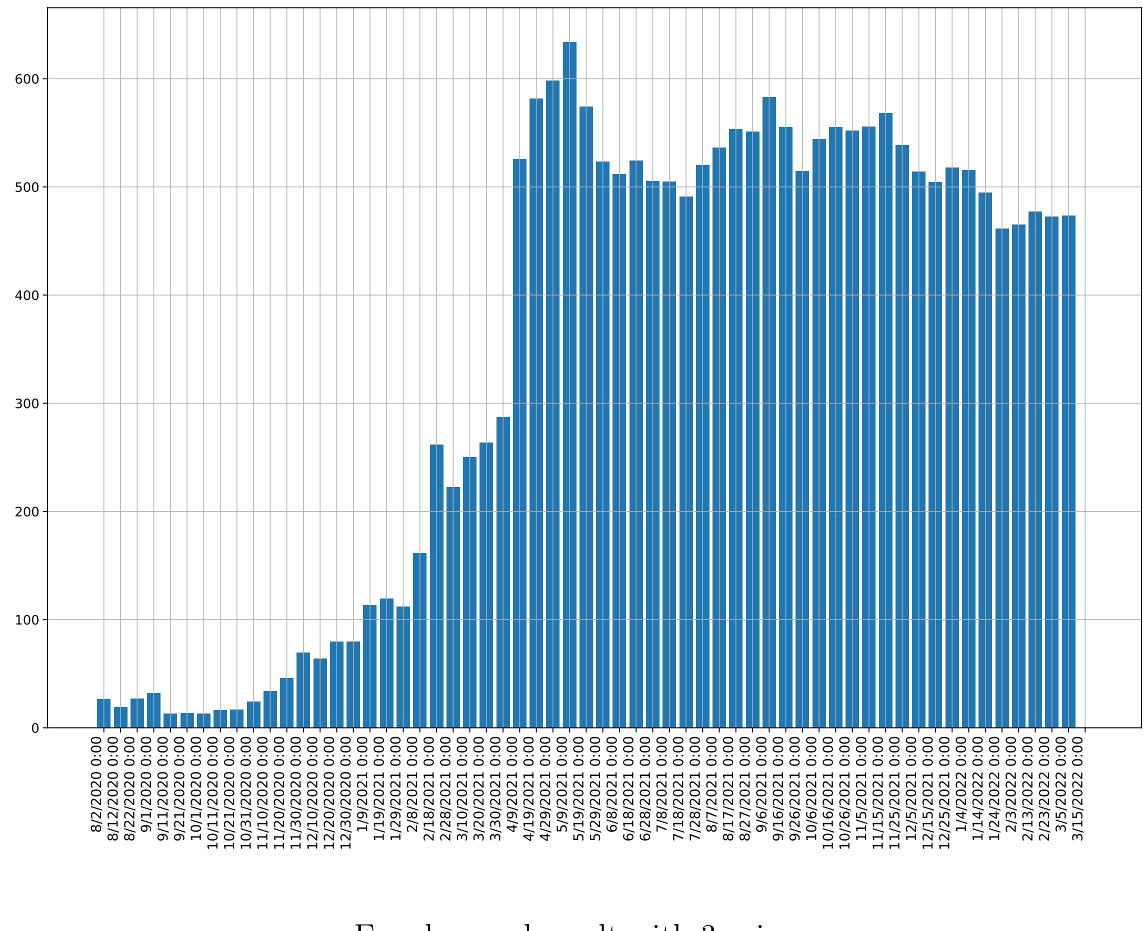
XRP results in profit percentage while using GRU prediction method.

B ES Strategy without BNB coin

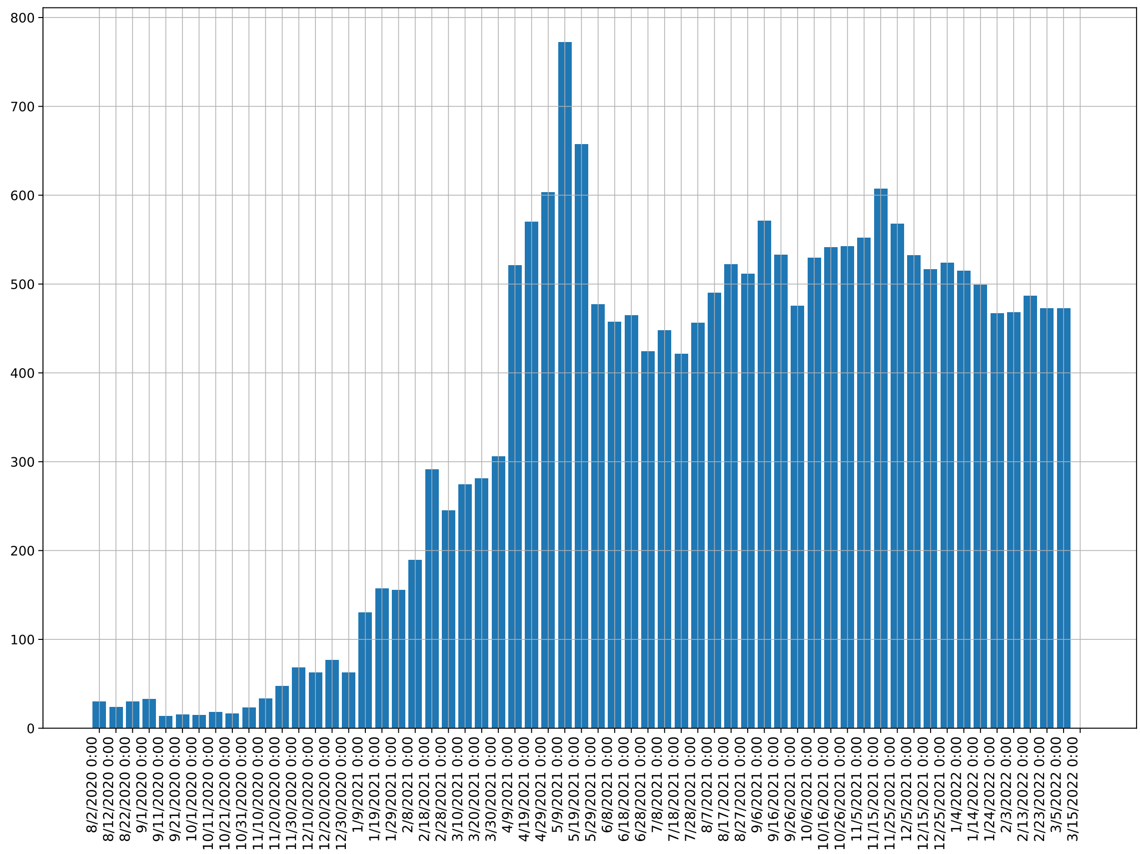
In this appendix we present results of ES strategy where we exclude BNB coin.



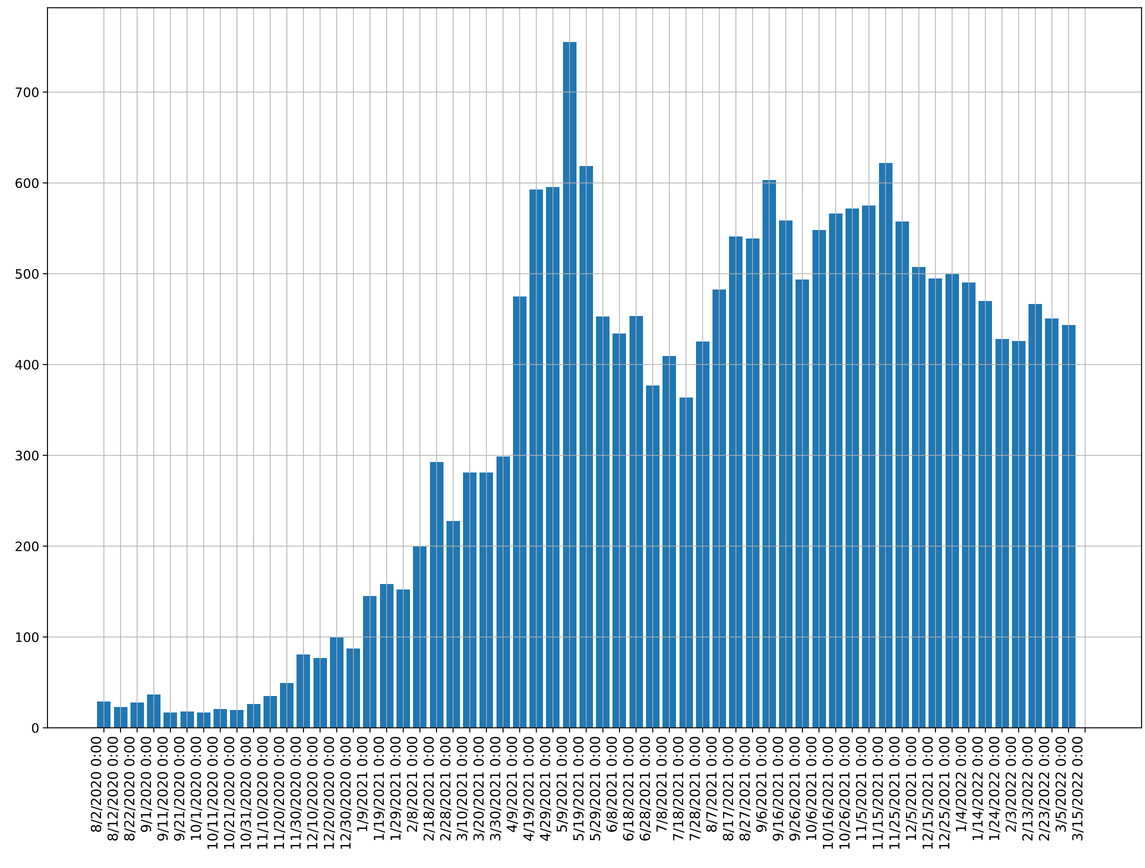
Equal spread result with 2 coins.



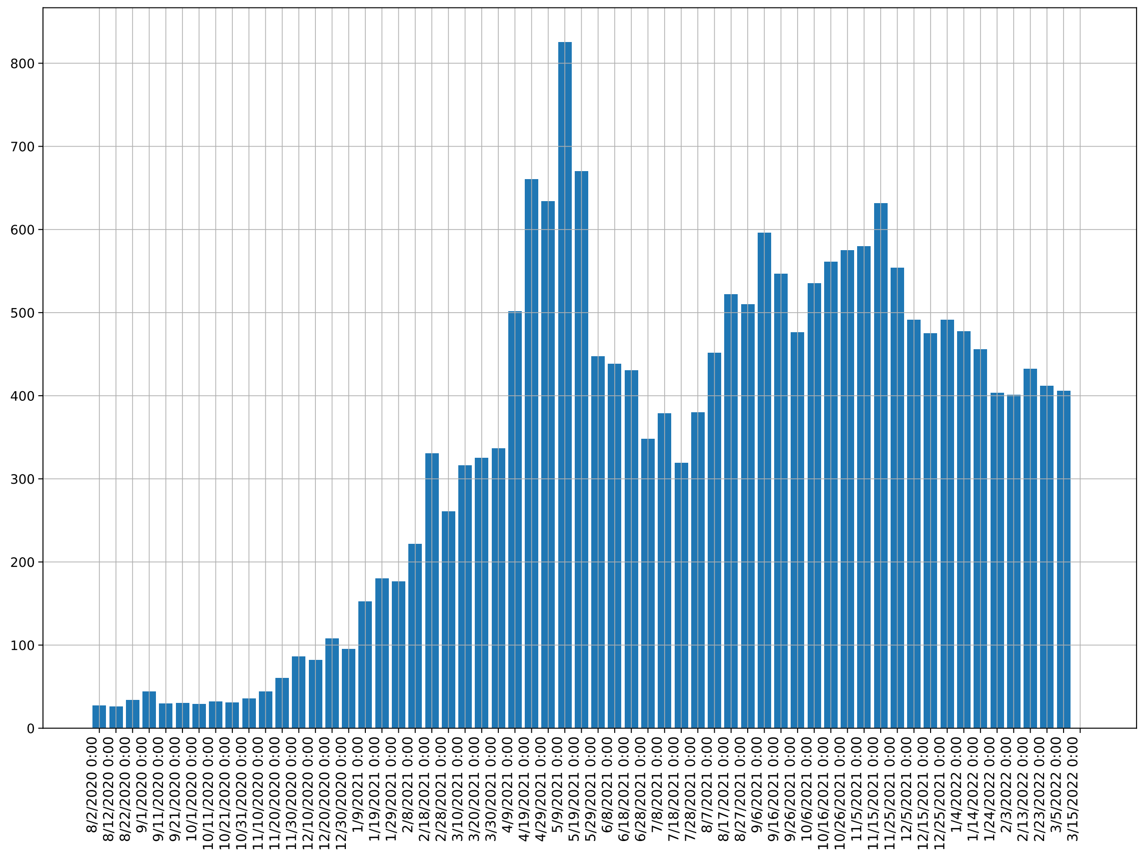
Equal spread result with 3 coins.



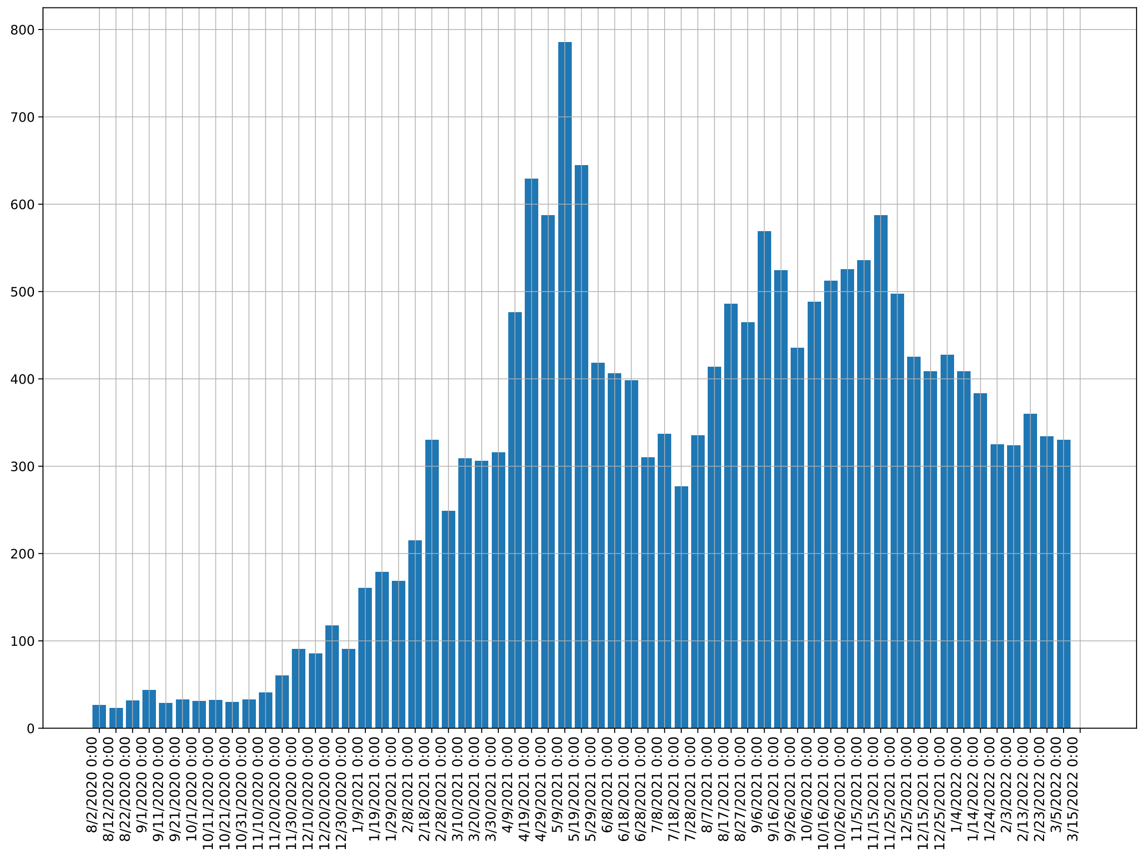
Equal spread result with 4 coins.



Equal spread result with 5 coins.



Equal spread result with 6 coins.



Equal spread result with 7 coins.

Here we see that by excluding most successful coin, we get quite different results.