Cryptocurrency Return Forecast via Machine Learning

Candidate Number: 5336J
MPhil in Economic Research

Abstract

This study establishes that the combination of technical analysis and deep learning networks can bring some level of predictability to cryptocurrency markets. Our analysis has a directional accuracy rate of 76.11% for Bitcoin, which is a 23% increase compared to the results of McNally (2018). We further conduct a range prediction, and the classification accuracy rate for Bitcoin is 20.56%. Also, we build up a trading strategy based on model predictions and backtest it from November 2018 to May 2019. We find Sharpe ratios and cumulative returns of the proposed trading strategy outperform the buy-and-hold benchmark for most cryptocurrencies. Finally, we conduct a feature importance analysis and find momentum indicators play the most important part in the model prediction process. Our analysis indicates the applicability of deep learning and technical analysis in cryptocurrency markets, and sheds light on the sentiments behind these volatile markets for researchers and investors.

Keywords: Deep learning, Technical analysis, Neural Network, Long short-term memory network, Recurrent neural network, Cryptocurrency, Return predictions, Feature importance analysis, Classification predictions

I acknowledge that this is my own work, and does not contain material that has already been used for a comparable purpose.

Word Limit: 10,000
Actural Work Count:9,950

1. Introduction

Cryptocurrency is a digital currency based on blockchain technology to increase security and transparency. It is designed as a stable exchange medium to facilitate transactions. However, the word 'cryptocurrency' nowadays is often associated with 'extreme volatility'. In addition, unlike stocks, the volatility of cryptocurrency usually cannot be well explained by changes in fundamentals. Therefore, it is natural to wonder, is it possible to forecast cryptocurrency returns? If so, how? This paper focuses on predicting cryptocurrency returns by combining deep learning networks with technical analysis.

Deep learning is a branch of supervised machine learning. Compared to traditional econometrics models, deep learning networks can handle far more input variables while allowing them to have non-linear interactions. We use three different deep learning networks to predict cryptocurrency returns, including a 3-layer feed-forward neural network, a 2-layer LSTM network, and a hybrid network that combines feed-forward layers and LSTM layers.

Our study shows that by combining deep learning networks with technical analysis, we can significantly improve the accuracy rate of cryptocurrency return predictions. In our 2-layer LSTM model, we have a bidirectional accuracy rate of 76.11% for Bitcoin, which is 23% higher than the results by McNally et al. (2018). We also expand our research focus to other five major cryptocurrencies and find an average bidirectional accuracy rate of 71.44%.

In addition to directional analysis, we further move to interval predictions and examine accuracy rates and win ratios across different networks and find an average accuracy rate of 20% using LSTM networks. Moreover, we build up a trading strategy based on model predictions and find Sharpe ratios of the proposed strategy based on LSTM networks are higher than or at least the same as the buy-and-hold benchmark for all cryptocurrencies. Finally, in order to understand the decision process behind the model, we conduct a feature importance analysis and find technical momentum indicators play the most important part in determining future returns.

The rest of the paper is organized as follows. Section 2 discusses the contributions and limitations of contemporary literature. Section 3 introduces different deep learning network methodologies. Section 4 discusses our model architecture design. Section 5 reports our empirical analysis results. Section 6 points out future research directions. Section 7 concludes.

2. Literature Review

This paper touches on the problem of applying machine learning models to cryptocurrency return predictions. While there are few research papers on this specific topic, the use of machine learning in the stock market is prevalent in recent years. Though the feasibility of stock price predictions has been a debated topic for decades, more and more empirical researches show that machine learning models could bring some level of predictability into stock price movements. Gu, Kelly and Liu (2018) provide a systemic study of machine learning application in stock pricing. They use various machine learning models, including random forest, neural network, elastic net and lasso regression, to predict stock risk premia and achieve high R^2 in out-of-sample return predictions. In a similar spirit, Feng et al. (2018) focus on 1-month asset return predictions and get positive R^2 in Neural Network models. They both attribute the predictive gain to the allowance of non-linear predictor interactions in machine learning models.

Compared to traditional econometric models, machine learning methods have the advantage of handling a large number of predictors while allowing them to have non-linear interactions. This makes them perfect tools for price predictions or return predictions in the financial market. However, unlike stocks and bonds, cryptocurrencies do not pay dividends or interests and thus have no underlying value. This means the existing asset pricing theory in the stock market will be an ill fit in predicting returns of cryptocurrencies. Vo and Xu (2016) fit the Bitcoin return series into a TGARCH model under the generalized hyperbolic distribution and then carry out a correlation analysis of Bitcoin returns and stock indices such as DJIA and ASX. They find correlation coefficients are less than 0.5 and conclude Bitcoin is not related to prices of other markets. Liu and Tsyvinski (2018) diver deeper into this topic by including two more cryptocurrencies, Ripple and Ethereum, into their study. They find all three cryptocurrencies not only have a small exposure to common risk factors in the stock market but also have little exposure to major currencies, precious metals, commodities and macroeconomic factors.

Furthermore, the volatility of the cryptocurrency market seems unprecedented. One Bitcoin only worth \$0.3 in 1st January 2011 and yet six years later its price is \$1033.3, which means around 344,300% rate of return over six years. The talk of bubbles is often heard in the cryptocurrency market. Geuder, Kinateder and Wagner (2018) use the PSY methodology proposed by Phillips et al. (2015) to detect bubbles. PSY methodology assumes asset prices show explosive behaviour during bubble periods and can identify potential bubbles with their beginning dates and end dates. In their study, the year 2016 and 2017 are characterized by multiple bubble periods.

The cryptocurrency market is unique in its sense. The lack of correlation to other markets, a small exposure to fundamentals, a few bubbles in the past periods, all these factors add to the difficulties of predicting cryptocurrency returns. The use of social media data is considered as an alternative method to predict cryptocurrency returns. Colianni et al. (2015) explore the possibility of predicting cryptocurrency returns using Twitter data. They use text-classification method to classify Twitter dataset with the different sentiment: positive, negative and neutral. They get an accuracy rate of around 90% in bidirectional return predictions. However, most of the drastic price movements in the Bitcoin market have not happened when this paper came out. Instead of using Twitter data, Phillips and Gorse (2017) use Reddit data and fit them into epidemic models, which in turn give them the probability of the existence of a bubble in a certain period. Then by adjusting trading thresholds, their analysis shows that the use of Reddit data can bring profitability and beat the buy-and-hold strategy.

Machine learning provides an alternative. Jang and Lee (2016) are one of the earliest to apply machine learning methods to Bitcoin price predictions. They use Blockchain information and macroeconomic factors to train and test a Bayesian Neural Network model. Their analysis shows that compared to linear regression benchmark, BNN model achieves lower Root Mean Square Error (RMSE) in predicting log price and log volatility. However, Liu et al. (2018) later show that there is little correlation between Bitcoin prices and macroeconomic factors so that their predictive power is limited. Besides, Jang and Lee (2016) do not offer any further evaluations on their models. Nor do they expand their research focus to cryptocurrencies other than Bitcoin.

Technical analysis seems to provide a solution to cryptocurrency's lack of correlation to other major financial markets. Detzel et al. (2018) introduce technical analysis into Bitcoin price predictions. They construct 5-, 10-, 20-, 50- and 100- day moving averages as prediction indicators and find statistically significant out-of-sample R^2. Then they build a trading strategy that long the Bitcoin if the price is lower than MAs and show this strategy could beat the buy-and-hold strategy. However, the rational equilibrium model they use involves many prior assumptions and beliefs. In addition, the structure of their model makes it hard to expand the number of technical indicators, which is crucial for technical analysis.

McNally et al. (2018) use Long Short-Term Memory (LSTM) network to predict the direction of Bitcoin prices and compare it to the ARIMA time series model benchmark. They use Bitcoin historical prices, Blockchain data and two simple moving averages as features. The highest bidirectional accuracy rate they find is 52% beating the accuracy rate of 50.05% by the ARIMA model. The LSTM model only wins by a narrow margin. However, it does not mean LSTM is a poor choice. Their unsatisfactory results can be caused by several reasons. Firstly, their limitation of features. Apart from Bitcoin historical prices and Blockchain data, they only include two simple moving averages as features. Even though they are engineered, it is still considered as far more insufficient for a proper technical analysis. Secondly, they stop to train the model if the validation loss does not improve for five epochs, which can also result in poor performance. Unlike feed-forward neural networks, LSTM takes more computation power and learning time. It is common for the LSTM model to improve validation loss after ten epochs of no progress. In other words, it learns slowly, but still, it learns.

Another attempt to use machine learning to predict cryptocurrency prices is by Alessandretti et al. (2019). They choose Boosting Trees, LSTM model and show that the LSTM model could better predict the prices when the window length is 50 days. In addition, they build a trading strategy based on their predictions and show it can bring positive profits after including up to 1% of the transaction cost. However, in their analysis, they use characteristics of 1,681 cryptocurrencies to predict each cryptocurrency price. Due to the short time frame of cryptocurrency, it is fair to say that their predictive features exceed their training datasets, and it can lead to many severe problems such as overfitting and instability of the model. Secondly, they do not compare their trading strategy to the buy-and-hold benchmark; therefore it is unclear whether their positive return comes from the superiority of their model or just from a general growing trend of the market.

Huang et al. (2018) are the first to combine thorough technical analysis with Bitcoin return predictions together. They firstly divide Bitcoin return into 21 intervals and process it using decision tree models. Their study shows that a trading strategy based on their model predictions can outperform the buy-and-hold strategy in terms of information ration. However, they do not expand the research focus to other cryptocurrencies and nor do they try to use features other than Bitcoin historical prices.

Therefore, to the best of our knowledge, none of these studies has systematically analyzed cryptocurrency returns by combining deep learning networks and technical analysis. With this paper, we aim to fill this void and make the following contributions to the literature.

Firstly, find an alternative way to predict financial assets which do not closely relate to the fundamentals and other financial markets. The technical analysis not only can be applied in cryptocurrency return predictions, but it can also be used to predict any financial instruments which have no acknowledged fundamentals.

Secondly, explore the applicability of different deep learning methods in cryptocurrency return predictions. In addition to traditional LSTM and Neural Network model, we also apply a new model proposed by Zhu et al. (2016), which is used to recognize human actions based on skeleton movements. We, therefore, denote it as the Skeleton model and it is a hybrid version of combining LSTM and feed-forward Neural Network model.

Thirdly, shed light on machine learning black box. We conduct a feature importance analysis to examine what features have the most predictive power in cryptocurrency return predictions. It will help researchers and financial traders to understand price movements of cryptocurrencies and conduct variable selections to increase training speed and model robustness.

3. Machine Learning in Finance

3.1 Feed-forward Neural Network

Artificial neural networks belong to the category of supervised learning and are algorithms that are inspired by the human brain to process complex information. When we see an image, millions of neural cells in our brain form an interactive web to process this information, generating electric signals and then pass them to the relevant area of the brain. Similarly, a neural network fits features into hidden layers and then compiles results into an output layer.

The most significant difference between feed-forward neural networks and recurrent neural networks is the ability to incorporate past output as features. For feed-forward networks, last period output will not act as features in the current period. Hence each period is treated independently. To train the model, we first need to specify a loss function. In this paper, we use mean squared error as our loss function, and its equation is defined as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{1}$$

In the training process, we use backward propagation algorithms to update optimal weights for each neuron. We use $f_1, \ldots, f_l$ to denote different hidden layers from 1 to L. In addition, each hidden layer has multiple neurons, and we use $f_1^{[1]}$ to denote the first neuron in the first hidden layer. To illustrate the training process behind, suppose we have a feed-forward network with one hidden layer which only contains one neuron. There are five input features and we denote it as $X \in \mathbb{R}^5$. The model structure is shown in figure 1. The first neuron will assign different weights and biases to each feature and then sum them up. Each neuron will pass the value to an activation function, which will conduct a non-linear transformation to neuron values and pass them to the next hidden layer. Most commonly used activation functions are sigmoid, tanh and rectified linear units (ReLU). In this paper, apart from classification models, we use ReLu as our activation value for each hidden layer. The equation of ReLU is defined as follows:

$$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases} \tag{2}$$

ReLu function can help to address vanishing gradient problems, for it is not bounded when x is positive, and it is more computationally efficient compared to exponential transformations by sigmoid functions.

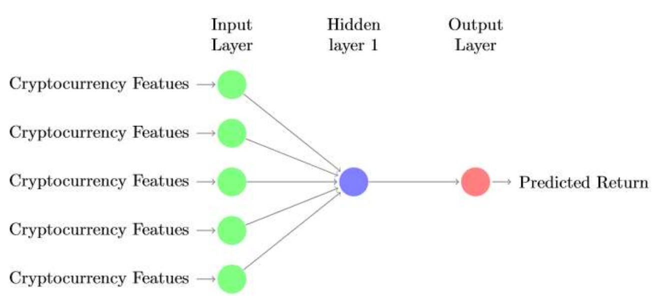

Figure 1**:** One-Layer Neural Network Structure

In this simple model, the output is calculated using the following equation:

$$f_1^{[1]} = g(\sum_{i=1}^{5} W_i X_i + b_i) \tag{3}$$

where W_i is weight for feature i, b_i is bias for feature i and g is an activation function.

By the same logic, for a neural network that has up to L hidden layer, its structure can be represented by the following equations:

$$R_t = F^{w,b}(X_t) \tag{4}$$

$$F^{w,b} = f_1^{w_1,b_1} \circ f_2^{w_2,b_2} \cdots \circ f^{w_L,b_L} \tag{5}$$

$$f^{w_l,b_l}(Z) = g(W_l Z + B_l) \tag{6}$$

where (w, b) are weights and biases to be trained by the model and g is an activation function that decides if the neuron is fired or not and what value is reported to the next layer.

3.2 Long Short-Term Memory Network

One severe drawback of feed-forward neural networks is their inability to use past output as features to train the model. The training process for feed-forward neural networks is independent for each input and thus, prior information cannot carry on to the next stage. To solve this problem, the idea of recurrent neural network has been developed. Recurrent neural networks can keep feeding past information into the current training process. However, plain recurrent neural networks also have one problem: the vanishing gradient. Hochreiter (1991) and Bengio et al. (1994) find, as the training process continues, the gradient of a plain recurrent neural network would gradually vanish to 0 or be too small to update the current weights. Therefore, it is difficult for a plain recurrent neural network to remember long term events.

To address the vanishing gradient problem, Hochreiter and Schmidhuber (1997) propose Long Short-Term Memory (LSTM) network. Each LSTM neuron has three gates: the input gate, the output gate and forget gate. By opening and shutting gates, LSTM could deliver the past information to the future stage intact. It is widely used in natural language processing, where the knowledge of the past helps to understand the meaning in the current context. In cryptocurrency return predictions, the ability to store past information is also important. Geuder, Kinateder and Wagner (2018) show that there were multiple bubble periods in cryptocurrency markets and we believe that by recognizing market trends and price patterns in the past could lead to better predictions of future returns.

The **s**tructure of a LSTM neuron is shown in Figure 2. We denote the information that stored in period t as C_t, output as h_t, and features as X_t. At time t, the past information C_{t-1}, the last period output h_{t-1} and features X_t are feed into the model.

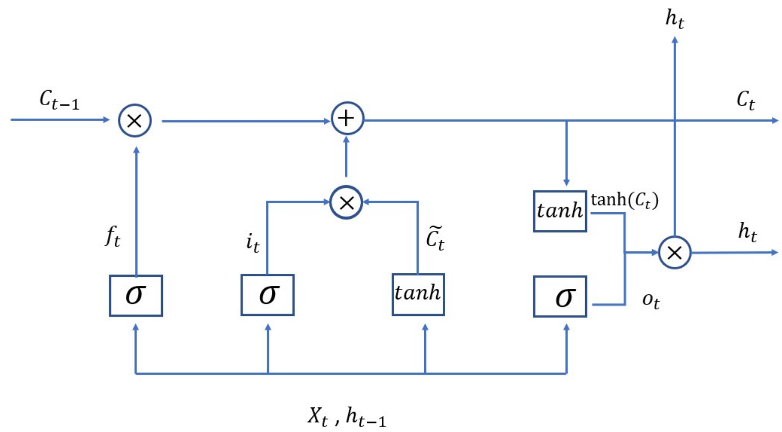

Figure 2: Long Short-Term Memory Neuron Structure

The forget gate is a sigmoid function and would decide what information to remember or to forget. At the beginning of each period, X_t and h_{t-1} are passed to the forget gate. It would transform the input to f_t within a $(0,1)$ interval, where 0 means completely forget and 1 means completely remember. Then the past memory C_{t-1} would operate a pointwise multiplication with the output of the forget gate.

The input gate is also a sigmoid function and decides what new information to update in the memory cell. At the beginning of the period, both X_t and h_{t-1} are passed to the input gate and it outputs a value i_t. Meanwhile, X_t and h_{t-1} are passed to a tanh function, which will transform the input into a new vector $\widetilde{C}_t$. Then i_t and $\widetilde{C}_t$ would have a pointwise multiplication.

The central part of LSTM model is to update the current information. This is a pointwise addition between input gate results and forget gate results. The new value C_t will be passed to the next stage and a tanh layer.

The output gate is a sigmoid function and decides what output to report in the current period. At the beginning of each period, both h_{t-1} and X_t go through the output gate and transform to a value o_t. Then the output is calculated by a pointwise multiplication between tanh (C_t) and o_t. The output value would be reported in this period and will continue to be features in the next period. The LSTM structure can be represented by the following equations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \tag{7}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \tag{8}$$

$$o_t = \sigma(W_o \cdot [ht-1, X_t] + b_o) \tag{9}$$

$$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, X_t] + b_c) \tag{10}$$

$$C_t = C_{t-1} \otimes f_t + \tilde{C}_t \otimes i_t \tag{11}$$

$$h_t = o_t \otimes tanh(C_t) \tag{12}$$

where σ represents a sigmoid function, $W_f, W_i,\ W_o$ are weight matrices for the forget gate, input gate and output gate respectively. i_t, f_t, o_t represents the input gate, the forget gate and output gate. C_t is the current state of a cell and h_t is the current output.

3.3 Hyperparameter Tuning

Machine learning is a flexible methodology and thus hyperparameter tuning is an essential part of the analysis. As we mentioned above, for the regression model, we use mean squared error as our loss function. For classification model, we use categorical cross entropy as our loss function. Its equation is shown as follows:

$$CE = -\sum_{i}^{C} t_i log(s_i) \tag{13}$$

where C represents the number of classification group and t_i, s_i are the groundtruth and the CNN score for each class.

Optimizer "Adam" is used to minimize the loss function. Adam is an algorithm first proposed by Kingma and Ba (2014). Different from traditional stochastic gradient descent (SGD), Adam uses the moving average of the gradient and can be treated as SGD with momentum. In addition, unlike SGD, which has the same learning rate for all parameters, Adam can adapt the learning rate for each parameter and works well across different neural network architectures, especially in non-stationary problems.

In each training epoch, we use mini-batch rather than the whole training sample. Batch size is the number of samples in the training data that the model would use to find the minimum gradient and update the model weights each time. Mini-batch gradient descent allows the model to update more frequently and is also more computationally efficient. Epoch is the number of times that the model completes training in the entire dataset. Figure 3 shows the relationship between model loss, batch size and epochs in a 3-layer neuron network model. We could see the smaller the batch size, the more volatile the loss. The mean squared loss using a batch size of 50 is smoother and lower than using a batch size of 10. In addition, increasing the number of epochs decreases model loss, but this impact gradually vanishes as the epoch number grows. After 100 epochs of training, there is little difference in the model loss using a batch number more than 10.

Dropout is another important machine learning regularization method. LSTM networks have the advantage of integrating past information into the current training process. Due to the same reason, LSTM networks easily face the risk of overfitting, especially when the number of the training sample is limited. To address this problem, a dropout layer is often used to force the network to ignore a part of nodes during the training process randomly. Adding dropout layers could allow the model to explore different potential relations between the input and output target. In addition, it can prevent the network from simply remembering the training set data.

In our analysis, we set epoch number to 100 and batch size to 32. To handle the potential problem of overfitting, we add a dropout layer after each LSTM layer and the dropout rate is set to be 20% across all three models. In addition, we use 20% of the training data to validate the model.

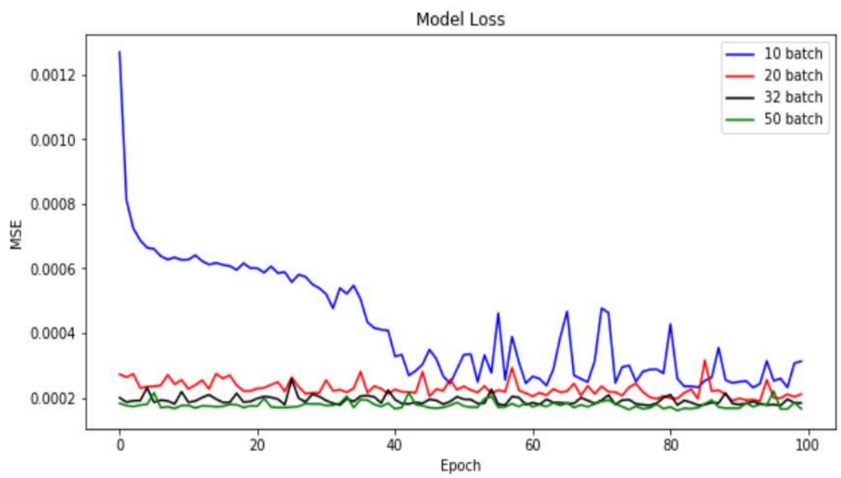

Figure 3: Model MSE with Different Batch Size and Epochs

4. Model Architecture

Model 1: The first model we use is a three-layer feed-forward neural network. The first hidden layer has 64 neurons, the second layer 64 neurons and the third layer 32 neurons. Figure 4 shows the structure of the model. We will refer to it as the NN3 model for the rest of the paper.

Let $R_t \in \mathbb{R}^{T\times 1}$ be a vector of cryptocurrency returns, $X_t \in \mathbb{R}^{T\times P}$ a high dimensional set of cryptocurrency predictors. This model can be represented as the following equations:

$$R_t = F^{w,b}(X_t) \tag{14}$$

$$F^{w,b} = f_1^{w_1,b_1} \circ f_2^{w_2,b_2} \circ f_3^{w_3,b_3} \tag{15}$$

$$f^{w_l,b_l}(Z) = ReLU(W_l Z + B_l) \tag{16}$$

where (w, b) are weights and biases to be trained by the model and ReLU is an activation function that only fires neurons whose values are above 0.

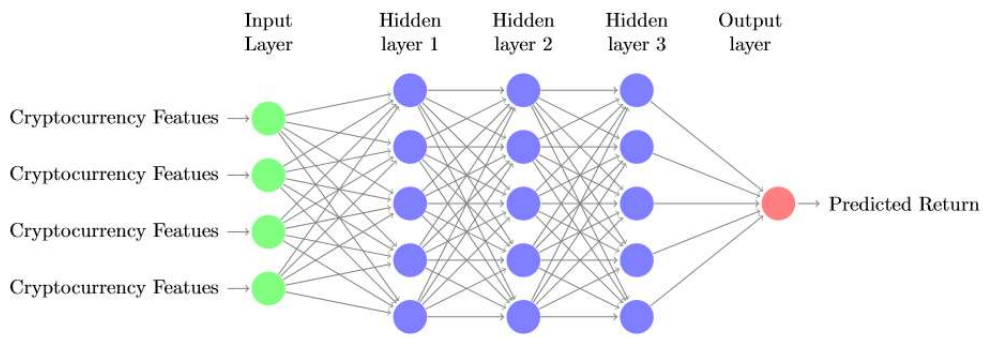

Figure 4: Three-layer feed-forward neural network structure

Model 2: The second model we use contains two LSTM layers, each having 32 neurons. The structure of the model can be illustrated in Figure 5. We will refer to it as the LSTM2 model for the rest of the paper. One thing worth noting is that the input dimension for LSTM networks is different from feed-forward neural networks. Training sets for LSTM networks have three dimensions (x, y, z), with x representing the number of the training sample, y representing the window length of each training sample and z represents the number of features in each training sample. For feed-forward neuron networks, training inputs only have two dimensions (x, z), with x presenting the number of training sample and z the number of features. The window length y is a special character of LSTM networks. It means in each training sample, apart from today's cryptocurrency characteristics, it also has the information of past y days. We coded as a flexible variable so that we can further examine the influence of the past information length on model performances.

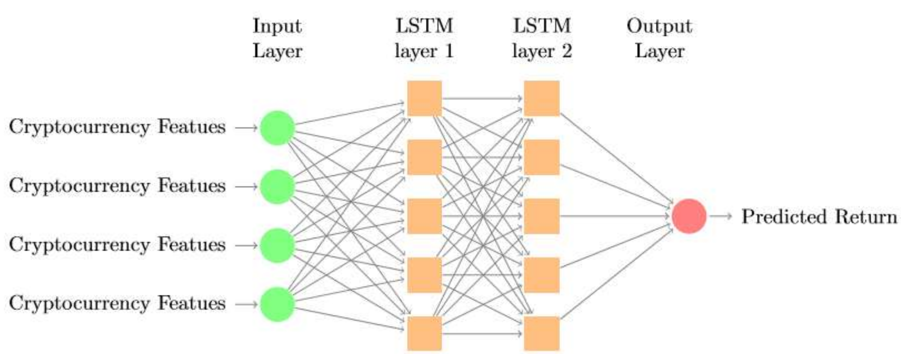

Figure 5: Two-layer long short-term memory network structure

Model 3: The third model we use contains five hidden layers. The first, third and the fifth layers are LSTM layers and the second and the fourth layers are feed-forward layers. The model structure is shown in figure 6. Each hidden layer has 32 neurons, and a drop layer is added after each LSTM layer. This model structure is originally proposed by Zhu et al. (2016) to recognize human actions based on skeleton and joint movements. Therefore, we will refer to it as the Skeleton model for the rest of the paper.

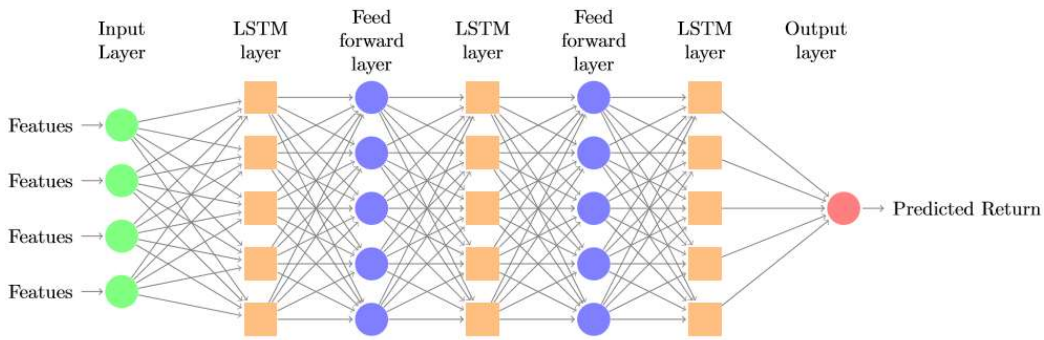

Figure 6: The skeleton model structure

5. Results

5.1 Data Description

Our study focuses on Bitcoin and five other major cryptocurrencies: Ethereum, EOS, Litecoin, Ripple, and Zcash. The data used in this analysis comprises three parts. Firstly, we get daily historical prices of cryptocurrencies from Yahoo Finance, which contains open, high, low, close, adj close prices and volume. The time range of each cryptocurrency differs due to their different issue date. For Bitcoin, the time frame starts from July 16th 2010; for Litecoin, it starts from October 23rd 2013; for Ripple it starts from January 21st 2015; for Ethereum it begins from 6th August 2015; for Zcash, it starts from 28th June 2017 and for EOS, the newest among the six, it starts from 24th September 2017.

Secondly, we also include daily Blockchain data from Quandl platform into our analysis. Blockchain data is a unique feature in the cryptocurrency market, and there are eleven variables we use in our study. They are Market Capitalization, Bitcoin My Wallet Transaction Volume, Bitcoin Average Block Size, Bitcoin Total Output Volume, Bitcoin Total Transaction Fees, Bitcoin Cost Per Transaction, Bitcoin Exchange Trade Volume, Bitcoin My Wallet Number of Transaction per day, Bitcoin Difficulty, Bitcoin Miners Revenue and Hash Rate.

Thirdly, to perform a technical analysis, we follow the method used by Huang et al. (2018) and generate 124 technical indicators using python TA library. Our technical indicators could be divided into five categories: overlap study indicators, momentum Indicators, volatility indicators, cycle indicators and pattern recognition Indicators. Full list of these indicators can be found in the appendix.

Overlap Study Indicators examine short-term past price movements and can identify a trend by smoothing volatile price movements. They are mostly lagging indicators, or trend-following indicators, including Bollinger bands, exponential moving average, Kaufman adaptive moving average, and so on.

Momentum Indicators measure the speed of price changes. The quicker the price rises, the more substantial the increase in momentum. They are mostly leading indicators, including the famous relative strength indicator (RSI) proposed by Wilder (1978), rate of change (ROC), balance of power, and so on.

Volatility Indicators, as the name implied, measure the degree of market volatility. The more volatile the price is, the larger the indicators. Variables in this category include the average true range proposed by Wilder (1978), true range and normalized average true range.

Cycle Indicators study cyclical fluctuations and create cycle period and phase. Indicators are purely mathematical and include various Hilbert transformations.

Pattern Recognition Indicators are different built-in price patterns, including Advance Block, Three Inside Up, On-Neck Pattern, and so on. If the price at a certain period can fit into a pattern, it will be shown as 1. Otherwise, it would be 0. Though pattern recognition indicators are crucial to technical analysis, many of them do not fit for a certain price movement. The value of these variables is 0 at all periods, and it is equivalent to feed noise into the training model. Therefore, we drop those ill-fit pattern recognition indicators to increase model stability.

5.2 Bidirectional Results

Firstly, we consider bidirectional return predictions of cryptocurrency. To evaluate different performances of the model, we use precision rate, recall rate, accuracy rate and F1 score. Evaluation statistics are calculated using the following equations:

$$Precision\ rate = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{17}$$

$$Recall\ rate = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{18}$$

$$Accuracy\ rate = \frac{True\ Positive + True\ Negative}{Total\ Number} \tag{19}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{20}$$

If the real return is positive and the predicted return is positive, then we define it as True Positive. If the real return is positive, but the predicted return is negative, then we denote it as False Positive. By the same logic, we also define True Negative and False Negative.

In a financial return prediction model, we will prefer a high recall rate to a high precision rate. Suppose we build a trading strategy based on bidirectional predictions and will long the cryptocurrency if the predicted return is positive and short if the predicted return is negative.

Hence if the number of False Positive is high, we will miss opportunities to make money, and if the number of False Negative is high, we would act according to wrong trading signals and suffer a financial loss. Investors are mostly risk-averse so that they would want to avoid False Negative more strongly. Therefore, ceteris paribus, the model with a higher recall rate is better than the model with a higher precision rate.

Table-1 reports statistic evaluations for bidirectional cryptocurrency return predictions using different models. For Bitcoin predictions, the LSTM2 model has a precision rate of 81.93%, an accuracy rate of 76.11%, a recall rate of 70.83% and a F1 score of 0.7598. For the Skeleton model, it reaches a precision rate of 88.46%, an accuracy rate of 68.89%, a recall rate of 47.92% and a F1 score of 0.6216. Among three networks, the NN3 model has the worst performance. It has a precision rate of 54.72%, an accuracy rate of 53.33%, a recall rate of 61.70% and a F1 score of 0.58.

Table-1: Bitcoin bidirectional results evaluation using different models

Name	**Model**	**TP**	**FP**	**TN**	**FN**	**Precision Rate**	**Accuracy Rate**	**Recall Rate**	**F1 Score**
	LSTM2	68	15	69	28	0.8193	0.7611	0.7083	0.7598
Bitcoin	Skeleton	46	6	78	50	0.8846	0.6889	0.4792	0.6216
	NN3	58	48	38	36	0.5472	0.5333	0.6170	0.5800
	LSTM2	83	38	49	10	0.6860	0.7333	0.8925	0.7757
Litecoin	Skeleton	88	51	36	5	0.6331	0.6889	0.9462	0.7586
	NN3	92	7	1	80	0.9293	0.5167	0.5349	0.6790
	LSTM2	88	40	45	7	0.6875	0.7389	0.9263	0.7892
Ripple	Skeleton	66	12	73	29	0.8462	0.7722	0.6947	0.7630
	NN3	95	13	0	72	0.8796	0.5278	0.5689	0.6909
	LSTM2	80	35	51	14	0.6957	0.7278	0.8511	0.7656
EOS	Skeleton	79	42	44	15	0.6529	0.6833	0.8404	0.7349
	NN3	74	47	20	39	0.6116	0.5222	0.6549	0.6325
	LSTM2	90	43	42	5	0.6767	0.7333	0.9474	0.7895
Ethereum	Skeleton	87	41	44	8	0.6797	0.7278	0.9158	0.7803
	NN3	89	29	6	56	0.7542	0.5278	0.6138	0.6768
	LSTM2	93	63	22	2	0.5962	0.6389	0.9789	0.7410
Zcash	Skeleton	92	59	26	3	0.6093	0.6556	0.9684	0.7480
	NN3	86	29	9	56	0.7478	0.5278	0.6056	0.6693

Note: This table calculates confusion matrix statistics for cryptocurrency predictions using different networks. TP stands for True Positive, FP stands for False Positive, TN stands for True Negative and FN is False Negative. The test sample starts from November 15th, 2018 to May 21st, 2019.

In addition, we could see technical analysis improves the performance of LSTM networks. McNally et al. (2018) have an accuracy rate of 52.78% for Bitcoin using LSTM networks. They use Bitcoin historical prices, blockchain data and simple moving averages. In our analysis, after adding 124 technical indicators, there is an 23.33% increase in the accuracy rate for the LSTM2 model and an 16.11% increase for the Skeleton model. This proves the applicability of technical analysis in the cryptocurrency market. By conducting a thorough technical analysis, we improve the accuracy rate of the LSTM network by around 20%.

For other cryptocurrencies, we could see the results are also consistent. Compared to the NN3 model, the LSTM2 model and the Skeleton model have a higher F1 score, accuracy rate and recall rate. Across five major cryptocurrencies, the LSTM2 model has an average F1 score of 0.77, the Skeleton model 0.76 and the NN3 model is 0.67. This shows the structure advantages of LSTM networks to store past information is useful in financial predictions.

To further analysis the bidirectional predictability of cryptocurrency, we apply Quantilogram test proposed by Linton and Whang (2004) to our cryptocurrency returns. The null hypothesis of this test is that there is no predictability in time series. Figure 7 gives the Bitcoin quantilogram for quantiles in the range 0.01 – 0.99 and out to 100 lags. In addition, 95% confidence intervals are shown by blue dashed lines. From figure 7, we could see there is some evidence to show predictability in Bitcoin, especially in quantiles between 75% to 90%. There are many observations outside the liberal confidence bands and a few outside the conservative bands. This can be shown more clearly by a Portmanteau test in Figure 8. To summarize, the results of these tests are consistent with our findings using LSTM networks: there is bidirectional predictability in cryptocurrency market. The portmanteau test and quantilogram for other cryptocurrencies could be found in the Appendix and they all suggest there is some level of bidirectional predictability in these markets.

As we mentioned above, inputs for LSTM networks have three dimensions (x,y,z). The window length y presents how many days of the past information is used in each training sample. Previously, the window length in LSTM networks is set to be 30 days, which means each training sample will remember the past 30 days information. However, different window length settings are influential to LSTM predictions. Therefore, we analyse the impact of past information by adjusting different window lengths. (Due to time constraint, we only focus on Bitcoin here.)

Table-2 presents evaluation statistics of the LSTM2 and Skeleton network using different window lengths. We could see from the results that different window lengths do have an impact on model performance. For the LSTM2 network, the precision rate increases as the window length increases. The precision rate is 81.82% when the window length is 10 days, 81.93% when the window length is 30 days and 87.5% when the window length is 100 days. There is a nearly 6% increase of precision rate by expanding window length from 10 days to 100 days. However, at the same time, the recall rate first increases and then declines as the window length becomes longer. The recall rate is 65.63% when the window length is 10-day-long, 70.83% when the window length is 30-day-long and sharply drops to 32.29% when the window length reaches to 50 days. The increase in precision rate and decrease in recall rate explains why the LSTM2 model has the highest F1 score when the window length is 30 days and has the lowest

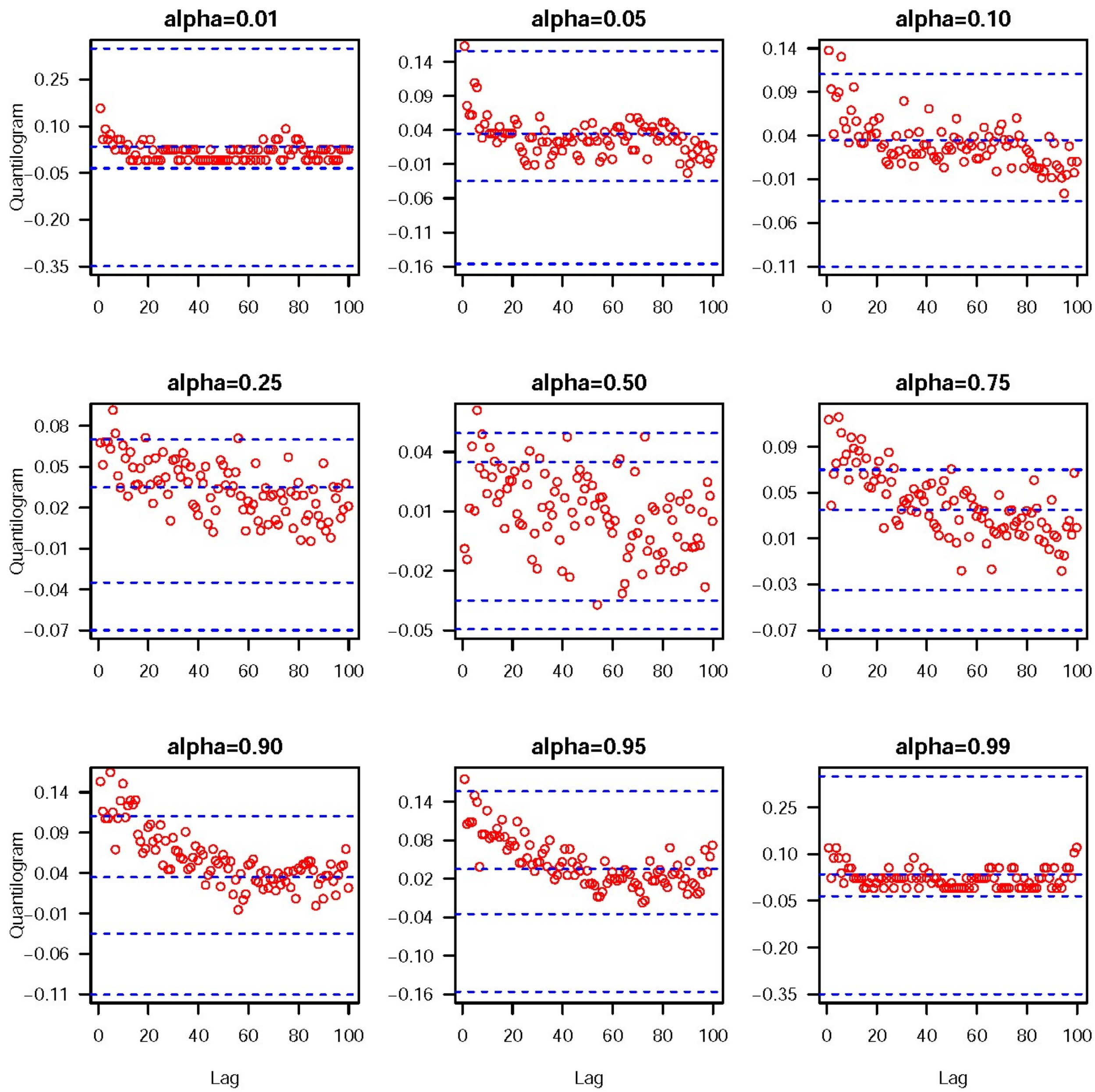

Figure 7: Quantilogram test for Bitcoin daily return

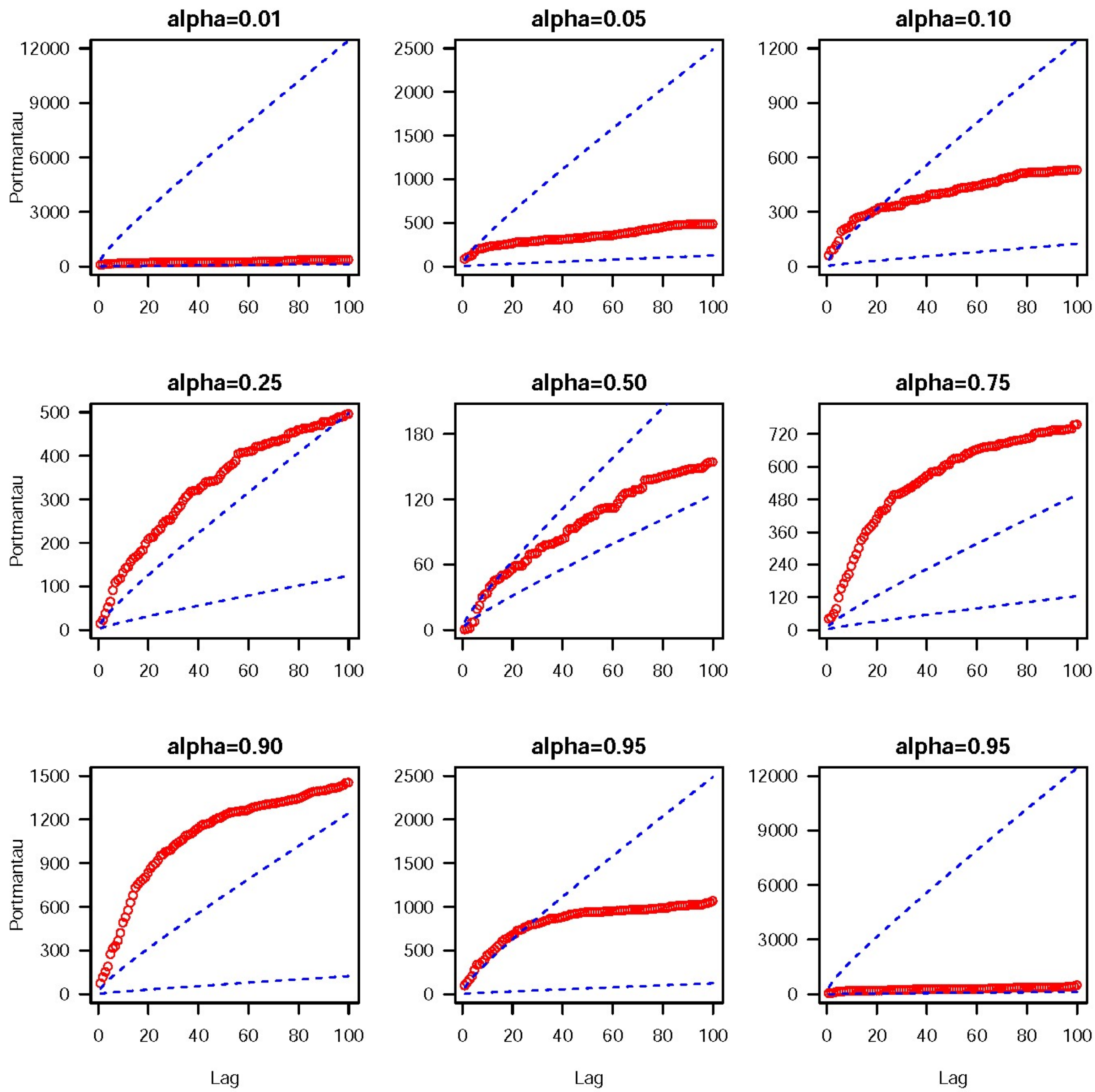

Figure 8: Portmanteau test for Bitcoin daily return

F1 score when the window length is 50 days. To summarize, the LSTM2 model has a similar F1 score with the window lengths from 10-day-long to 30-day-long, all of them in a 5% up-or-down interval around 0.75. However, when the window length extends, there is a sharp drop in F1 score, from an average of 0.74 to 0.54. In addition, the accuracy rate drops when the window length becomes longer. Therefore, the LSTM2 model is more suitable for short-term past information storage.

Table-2: The impact of window length on model performances

Model	Window Length	TP	FP	TN	FN	Precision Rate	Accuracy Rate	Recall Rate	F1 Score
LSTM2	10	63	14	70	33	0.8182	0.7389	0.6563	0.7283
	20	68	21	63	28	0.7640	0.7278	0.7083	0.7351
	30	68	15	69	28	0.8193	0.7611	0.7083	0.7598
	40	35	3	81	61	0.9211	0.6444	0.3646	0.5224
	50	31	5	79	65	0.8611	0.6111	0.3229	0.4697
	100	42	6	78	54	0.8750	0.6667	0.4375	0.5833
	180	42	7	77	54	0.8571	0.6611	0.4375	0.5793
Skeleton	10	90	41	43	6	0.6870	0.7389	0.9375	0.7930
	20	50	8	76	46	0.8621	0.7000	0.5208	0.6494
	30	46	7	77	50	0.8679	0.6833	0.4792	0.6174
	40	73	16	68	23	0.8202	0.7833	0.7604	0.7892
	50	89	48	38	5	0.6496	0.7056	0.9468	0.7706
	100	57	11	73	39	0.8382	0.7222	0.5938	0.6951
	180	75	18	66	21	0.8065	0.7833	0.7813	0.7937

Note: This table calculates confusion matrix statistics for cryptocurrency predictions using different networks. TP stands for True Positive, FP stands for False Positive, TN stands for True Negative and FN is False Negative. The test sample for different window length has different start dates and end dates but the total test samples are 180 days.

The Skeleton model, on the other hand, is the opposite. For the skeleton model, the precision rate also increases when the window length expands. It has a precision rate of 68.70% when the window length is 10-day-long and then increases to 83.82% when the window length reaches 100 days. However, unlike the LSTM2 model, the recall rate of the Skeleton model also improves as the window length increases. The Skeleton model recall rate is only 52.08% when using 20-day-long window, yet it achieves 78.13% when window length expands to 180 days. Hence the Skeleton model has the highest F1 score when the window length is 180 days. One thing worth noting about the Skeleton model is that it is also suitable for both extremely short and long window length. The F1 scores it gets for the 10-day-long window length and the 180-day-long window length only differs by 0.0007. Therefore, its ability to process short-term past information and long-term past information exceeds the LSTM2 model and the performance of the Skeleton model is also more stable.

5.3 Classification Results

Directional predictions usually are not enough in the financial market. Besides the direction of the price, we also want to know exactly what amount of increase or decline there will be in the future. To predict more than directions, we firstly use classification models in deep learning. Classification models can output results into different categories based on their features and are widely used in graph recognition problems.

To prepare the data into categorical data, we adopt the classification rule proposed by Huang et al. (2018) and divided daily return into 21 non-overlap intervals. Each interval is represented by an integer from -10 to 10, with 0 representing daily return from -0.2% to +0.2%, 1 representing +0.2% to +0.4%, -1 representing -0.2% to -0.4% and so on so forth. Any daily return higher than 2% is represented by 10 and any daily loss bigger than -2% is represented by -10. If the predicted range is the same as the real range, we denote it as a True range prediction. Otherwise, it is a False range prediction. The accuracy rate is calculated the using the following equation:

$$Accuracy\ Rate = \frac{True\ range\ predictions}{Total\ number\ of\ predictions} \tag{21}$$

Apart from classification models, regression models are also being used to calculate the accuracy rate. One significant difference between regression models and classification models is that regression models would directly output the exact value of the predicted return while classification models will out possibilities for each interval. For regression models, we classify their predicted returns into the above 21 intervals to get range predictions. For classification models, we take the interval with the maximal possibility as our predicted range. In addition, the categorical cross-entropy loss function is used to train classification models, while the mean squared error loss function is used to train regression models. Table-3 presents the results of cryptocurrency range predictions.

From Table-3, we could see the accuracy rates of regression models are significantly higher than classification models. For Bitcoin, it achieves an accuracy rate of 20.56% using the LSTM2 regression model, but only has an accuracy rate of 8.89% when using the classification model. For Litecoin, the accuracy rates using classification models are around 2.5% while it reaches to 35.56% using regression models. This also applies to all other cryptocurrencies. Accuracy rates using regression models are invariably higher than using classification models. One explanation for the underperformance of classification models is information loss. When we transform the original daily return into categorical data, a part of return information is lost.

Table-3 Cryptocurrency Range Predictions

Cryptocurrency	Model	Classification Accuracy Rate	Regression Accuracy Rate
	LSTM2	0.0889	0.2056
Bitcoin	Skeleton	0.0778	0.0444
	NN3	0.0833	0.2278
	LSTM2	0.0222	0.3556
Litecoin	Skeleton	0.0278	0.3556
	NN3	0.0222	0.3556
	LSTM2	0.0389	0.1333
Ripple	Skeleton	0.0444	0.1389
	NN3	0.0556	0.2000
	LSTM2	0.0223	0.2167
EOS	Skeleton	0.0223	0.2778
	NN3	0.0223	0.2722
	LSTM2	0.0833	0.2556
Ethereum	Skeleton	0.0778	0.1889
	NN3	0.0722	0.2667
	LSTM2	0.0333	0.3111
Zcash	Skeleton	0.0444	0.2278
	NN3	0.0333	0.2167

In addition, compared to directional accuracy rates, range accuracy rates drop greatly. The LSTM2 model has an average accuracy rate of over 70% in bidirectional predictions while has an average accuracy rate of around 24% in range predictions. This is understandable since the prediction range is much more difficult. However, except for the Skeleton model for Bitcoin, we found there is little difference between LSTM networks and feed-forward networks. In other words, the ability to store past information does not give many advantages to LSTM networks in range predictions. For Bitcoin, Ethereum and Ripple, the NN3 network achieves the highest accuracy rate, all around 20%. Besides the NN3 model gets an accuracy rate of 35.56% for Litecoin, the same as the LSTM2 model and the Skeleton model.

One explanation for the underperformance of LSTM networks in range prediction is too much noise in the past information. LSTM can store past information, but not all of them is useful in predicting future returns. Too much stored information complicates the model structure. In addition, due to a short history in cryptocurrencies, we are unable to use a large amount of data to exploit the potential structural benefits of LSTM networks fully. Therefore, the imbalance of model complications and limited sample observations lead to a slight underperformance of LSTM networks compared to the feed-forward Neural Network in range predictions.

To further evaluate range predictions in each interval, we follow the method proposed by Huang et al. (2018) and construct a win ratio for each interval. Suppose an investor build a strategy based on the range predictions. If the predicted range is +3, then the investor would long the cryptocurrency, and if the following day return is positive then the investor makes money. Otherwise, the investor suffers a loss. By the same logic, if the predicted range is -3, then the investor would short the cryptocurrency and if the following day return is negative, the investor makes a profit. We defined these predictions that lead to profits for investors as profitable predictions for a certain range. Win ratio is calculated using the following equations.

$$Win\ Ratio = \frac{Profitable\ range\ predictions}{Frequency\ of\ the\ range} \tag{22}$$

Table-4 reports the win ratio for the 21 non-overlap range using the LSTM2 model. From the table, we could see the win ratio for the LSTM2 model ranges from 0 to 1.

For the LSTM2 model, we could see range +3 (0.6%: 0.8%) and range -3 (-0.6%: -0.8%)) have win ratio higher than 50% for all cryptocurrencies. For Bitcoin, the LSTM2 model has a win ratio of 100% for range -5 (-1.2%: -1.4%) and range 7 (1.6%: 1.8%). In addition, for Litecoin and Zcash, the LSTM2 model only predicts large positive ranges, which implies the LSTM2 model is better at detecting larger return movements. The table reports prediction range win ratios using other two models can be found in the Appendix. We could see from the table that the NN3 model predictions is more sensitive to small ranges. It reports eight different ranges for Litecoin while both the LSTM2 model and the Skeleton model only report +10 range predictions. However, for the NN3 model, the increasing coverage of range predictions do not increase its win ratio. Figure 9 to figure 14 show the predicted returns and the real return using the LSTM2 model and graphs using the other two models can be found in the Appendix.

In figure 9 to figure 14, the real return is represented by the red line and the predicted return is presented by the green line. The LSTM2 model acts like a trend filter, which captures the average price movements well. We could see that, for all cryptocurrencies, the real return is smoother than the real return. In addition, the LSTM2 model tends to report positive return predictions, which is illustrated well in Table-4. For Zcash, we could see return predictions mostly are above the real returns. This implies the LSTM2 model is slightly over-optimistic about the market. One drawback for the LSTM2 model is that it cannot predict extreme return movements. From the graphs, we could see both positive and negative extreme return movements are rarely captured by the LSTM2 model. Therefore, to improve the ability of the LSTM2 model to predict large returns is a future direction to improve model performance.

Table-4 Win Ratio of LSTM2 Model

Return Range	Bitcoin	Litecoin	Ripple	Zcash	Ethereum	EOS
-10	0.5714	-	-	-	0.0000	0.5714
-9	0.0000	-	-	-	-	0.0000
-8	0.0000	-	-	-	0.0000	0.0000
-7	-	-	-	-	0.6667	-
-6	-	-	-	-	0.0000	-
-5	1.0000	-	-	-	0.5000	1.0000
-4	0.0000	-	-	-	0.8000	0.0000
-3	0.6667	-	-	-	0.7500	0.6667
-2	0.2000	-	-	-	0.3333	0.2000
-1	0.4545	-	-	-	0.3750	0.4545
0	0.4000	-	0.0000	-	0.5833	0.4000
1	0.4118	-	-	-	1.0000	0.4118
2	0.4444	-	0.3333	-	0.1667	0.4444
3	0.5500	-	0.7500	-	0.5000	0.5500
4	0.3250	-	0.5000	-	0.5714	0.3250
5	0.2500	-	0.2500	-	0.5714	0.2500
6	-	-	0.4667	-	0.5000	-
7	1.0000	-	0.4706	-	0.3333	1.0000
8	-	-	0.6364	0.0000	0.5000	-
9	-	-	0.6154	-	0.2222	-
10	-	0.4778	0.5093	0.5196	0.4521	-

Note: This table provides win ratios of the LSTM2 model range predictions. The 21 non-overlap range are indexed by a number from -10 to 10. Win ratio is calculated by the number of true predictions over the frequency of the predicted range and "-" represents missing value because this range is not predicted by the LSTM2 model. The test sample period is from November 15th, 2018 to May 21st, 2019.

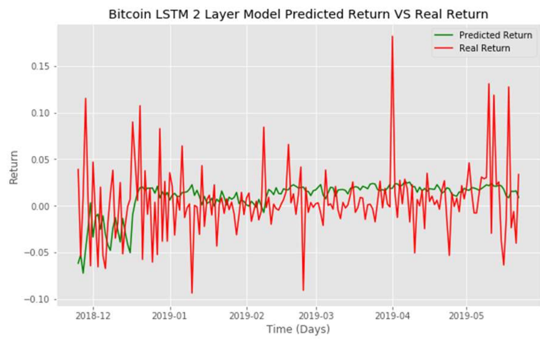

Figure 9: Bitcoin LSTM2 Predicted Return

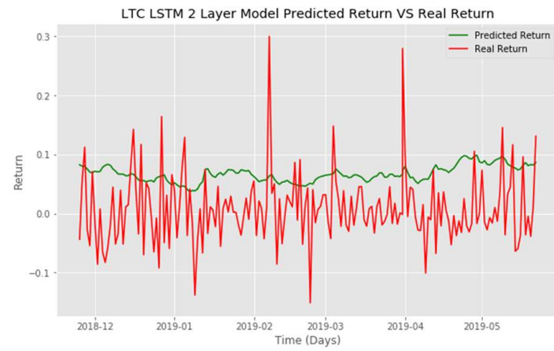

Figure 10: Litecoin LSTM2 Predicted Return

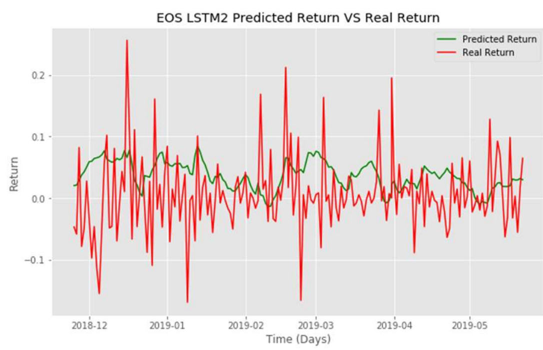

Figure 11: EOS LSTM2 Predicted Return

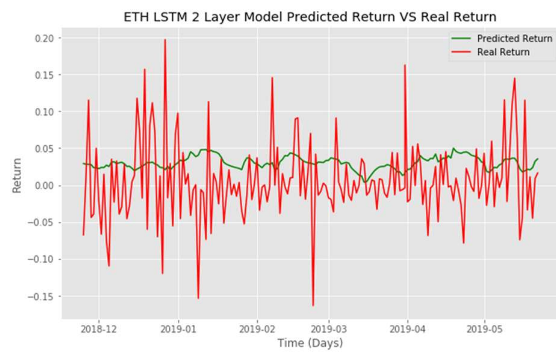

Figure 12: Ethereum LSTM2 Predicted Return

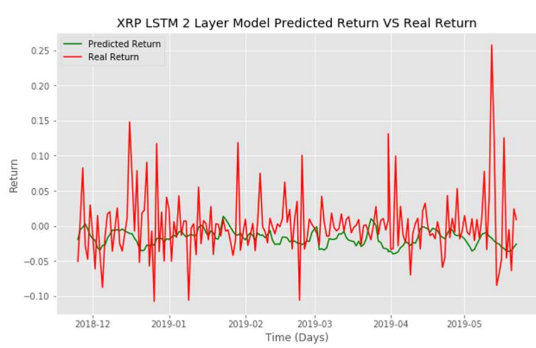

Figure 13: Ripple LSTM2 Predicted Return

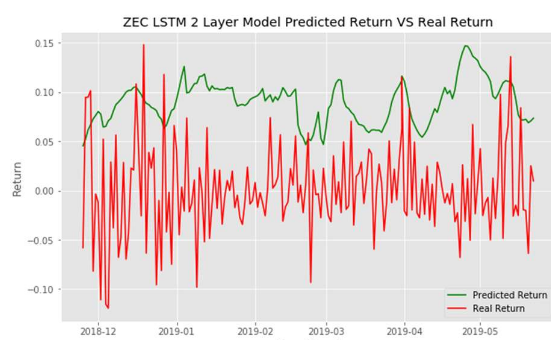

Figure 14: Zcach LSTM2 Predicted Return

5.4 *Backtesting Trading Strategy*

In the above section, we show the LSTM2, Skeleton and the NN3 model could predict directions better than specific return intervals. Therefore, in this section, we construct a trading strategy based on the bidirectional predictions of our analysis. Suppose an investor holds a wealth of one unit at the beginning. The investor would long the cryptocurrency if the model predicts the next day return will be in the range between +1 to +10. He would short the cryptocurrency if the next day return prediction is in the range between -1 to -10. If the next day model prediction range is 0, then the investor will not trade and will remain his position in the last period. In addition, we ignore transaction costs.

Table-5 reports Sharpe ratios of backtesting our trading strategy from November 15th, 2018 to May 21st, 2019. Sharpe ratio is developed by William Sharpe and it is equivalent to risk-adjusted return. The ratio is calculated using the following equations:

$$Sharpe\ \ Ratio = \frac{R_p - R_f}{\sigma_p} \tag{23}$$

where R_p is the return of the portfolio, R_f is the risk-free rate and σ_p is the standard deviation of the portfolio. In our analysis, we set the daily risk-free rate to 0.001. From Table-5, we could see for the buy-and-hold strategy, all Sharpe ratios are statistically different from 0 at 99% confidence level. Bitcoin, Litecoin, EOS, Ethereum bring positive returns while Zcash and Ripple bring slight loss to investors. Compared to the buy-and-hold strategy, Sharpe ratios of the proposed trading strategy based on the Skeleton model are invariably higher or at least the same. The Skeleton model brings positive returns for all cryptocurrencies. We also test the significance level of Sharp ratios for the proposed trading strategy against the buy-and-hold strategy, using chi-squared statistics. However, none of them is statistically significant. Only the Sharpe ratio between the NN3 and the buy-and-hold strategies in EOS are statistically different at 10% significance level.

Table-5 Sharpe Ratio of Trading Strategies

Name	**Strategy**			
	LSTM2	**NN3**	**Skeleton**	**buy-and-hold**
Bitcoin	1.17*	0.90*	1.06*	1.06*
Litecoin	1.49*	1.49*	1.49*	1.49*
EOS	1.12*	-1.26*	0.97*	0.78*
Ethereum	1.04*	-0.08	1.05*	1.05*
Zcash	-0.23*	-0.02	0.20*	-0.30*
Ripple	-0.50*	-0.22*	0.26*	-0.32*

Note: This table calculates Sharpe ratios of the proposed trading strategy based on different model predictions. We test the Sharpe ratio against 0 benchmark using t-test, and * suggest the value is significant at 99% confidence level.

Among all three strategies, the LSTM2 model and the Skeleton model have better performances. This is consistent with our above analysis that LSTM networks have a structural advantage over simple feed-forward networks.

For Bitcoin, the highest Sharpe ratio is 1.17 using the LSTM2 model and the NN3 model has the lowest Sharpe ratio of 0.9. For Litecoin, the Sharpe ratio for all strategies is the same. This is caused by the over-optimism of networks. All the LSTM2, NN3 and Skeleton model report positive range for Litecoin in all test periods. Therefore, this leads to an equivalence of using the buy-and-hold strategy based on model predictions. For EOS, the LSTM2 model has the highest Sharpe ratio 1.12, the Skeleton model the second, 0.97, the buy-and-hold strategy the third, 0.78, while the NN3 model has a Sharpe ratio of -1.26. This means the trading strategy based on the NN3 model predictions would bring negative returns to the investor. It implies the NN3 model predictions for EOS has a higher recall rate and offsets the profits brought by correct predictions. For Ethereum, the Skeleton model has the same Sharpe ratio as the buy-and-hold strategy, for it reports positive ranges in all periods and suggests the investor to hold the cryptocurrency during all test periods. For Zcash and Ripple, the buy-and-hold strategy would bring negative profits to investors during the test periods. Among our models, if the underlying asset brings negative profits, only the Skeleton model could bring positive returns to the investor. The LSTM2 model and the NN3 model all bring negative returns to investors even though their risk-adjusted returns are slightly higher than the buy-and-hold benchmark.

To summarize, trading strategies based on LSTM networks perform better than feed-forward networks. Sharpe ratio based on the Skeleton model outperforms the buy-and-hold strategies for EOS, Zcash and Ripple. As for Bitcoin, Litecoin, and Ethereum, the Skeleton model suggests the same strategy as the buy-and-hold benchmark for investors. Moreover, all Sharpe ratio using the Skeleton model are statistically different from 0 at 99% confidence level. However, it is difficult for the LSTM networks to beat the buy-and-hold strategies. Even though LSTM networks have a higher Sharpe ratio for various cryptocurrencies than the buy-and-hold benchmark, none of them is statistically significant. Finally, Sharpe ratios of the proposed trading strategy are closely related to the price movements of the underlying asset. For Zcash and Ripple, where the buy-and-hold strategies would bring negative returns to investors, the NN3 and the LSTM2 model all bring negative returns to investors as well. Even though the Skeleton model manages to bring positive returns to investors, the Sharpe ratios of these two cryptocurrencies are lower than other assets.

Figure 15 to figure 20 show the cumulative return of the buy-and-hold strategy, which is denoted as the black line, and of strategies based on the Skeleton model predictions, which is denoted as the red line. Cumulative return graphs using the LSTM2 model and the NN3 model can be found in the appendix.

Figure 15: Cumulative Return for EOS

Figure 16: Cumulative Return for Ethereum

Figure 17: Cumulative Return for Litecoin

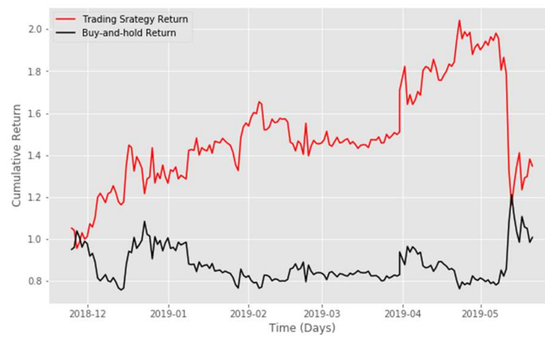

Figure 18: Cumulative Return for Ripple

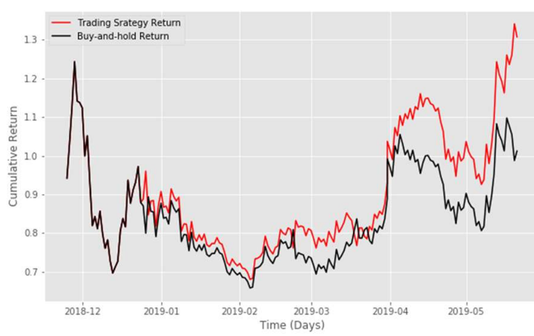

Figure 19: Cumulative Return for Zcash

Figure 20: Cumulative Return for Bitcoin

From these graphs, we could see for Zcash, EOS and Ripple the cumulative return using the Skeleton model is higher than the buy-and-hold strategy in all periods. All of them bring positive returns. During the test period, the strategy built on the Skeleton model predictions could bring around 200% returns for investors investing in EOS, around 210% for investors in Ethereum, nearly 300% in Litecoin, around 140% for investors in Ripple, 130% for investors in Zcash and around 200% for investors in Bitcoin.

These return rates are impressively high, but we also need to know the market in this period undergoes a huge rally. The price of Bitcoin nearly doubled during this period. Figure 21 analyze the correlation between cryptocurrency and the other five major cryptocurrencies. We could see they are highly positively correlated. For Litecoin and Ethereum, whose correlation with Bitcoin are around 90% both undergo a large market rally during the test period.

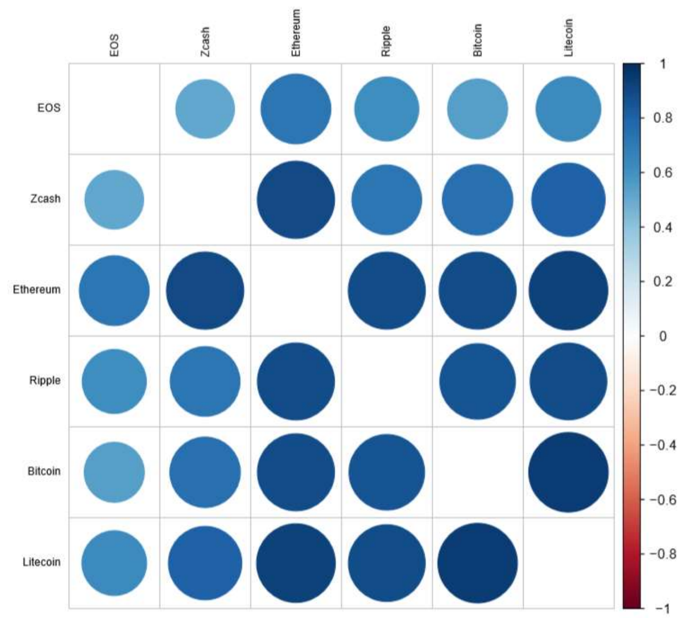

Figure 21: Correlation Analysis cross Cryptocurrencies

5.5 Feature Importance

Machine learning process sometimes is called as a black box, for we know little things happened in between. However, in addition to the results, features importance is also an essential part, for it can shed lights on the model and add interpretability to the results. Firstly, we conduct a correlation analysis between our output targets and each predictive indicator we use. Figure 22 shows the correlation analysis in Bitcoin. From the graph, we could see, all features are positively correlated to returns. Among them, normalized average true range (NATR) has the closest relation. Correlation analysis graphs for other cryptocurrencies can be found in the appendix.

We use Xgboost algorithm to get feature importance score. Xgboost construct gradient boosting decision trees and optimize the value of the objective function, in our case, the mean squared error. In a single decision tree T, Breiman et al. (1984) proposed to calculate the importance score for each predictor feature X_l using the following equations:

$$w_l^2(T) = \sum_{t=1}^{J-1} \hat{\tau}_t^2 \tag{24}$$

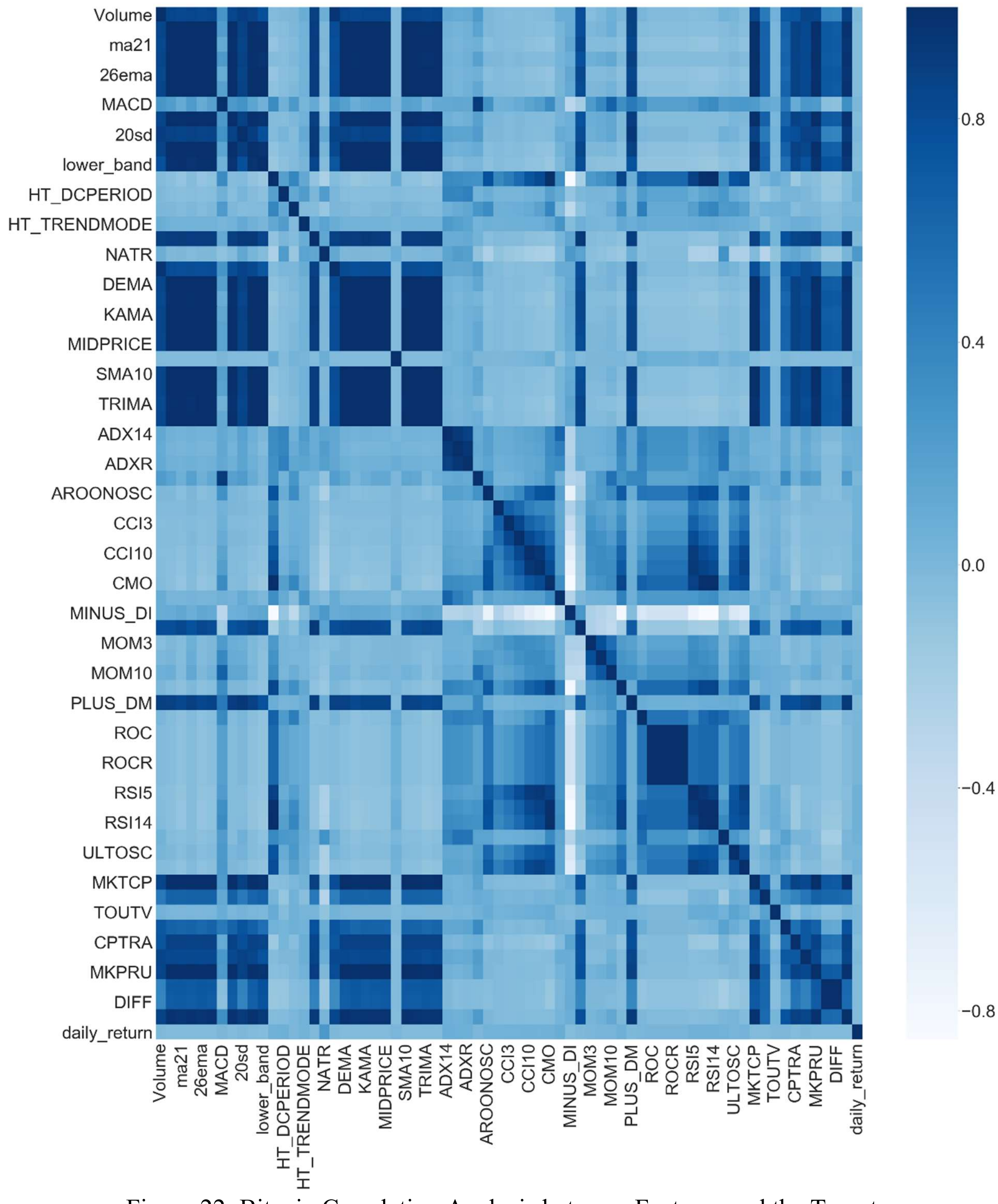

Figure 22: Bitcoin Correlation Analysis between Features and the Target

where J is the number of internal nodes in the decision tree and $\hat{\tau}_t^2$ is the estimated improvements by feature X_l. For M trees, the importance calculation equation is shown as follows:

$$w_l^2(T) = \frac{1}{M} \sum_{m=1}^{M} \hat{\tau}_t^2(T_m) \tag{25}$$

The following graphs show the 20 most important features for each cryptocurrency.

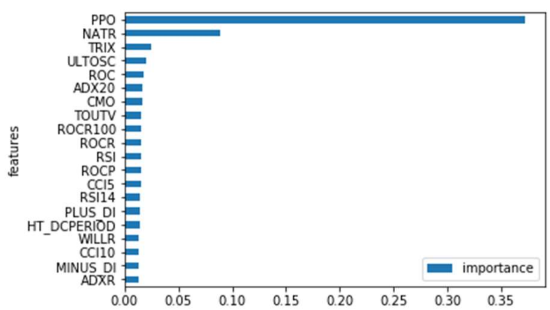

Figure 23: Bitcoin Feature Importance

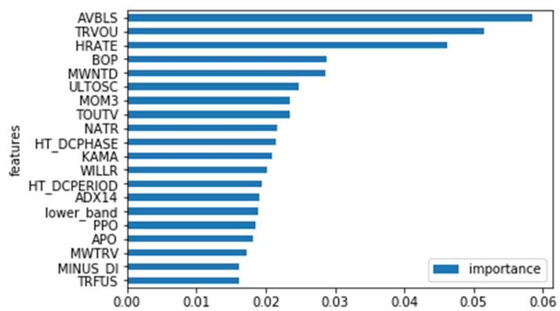

Figure 24: EOS Feature Importance

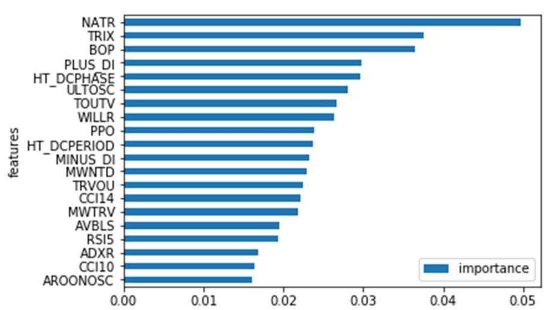

Figure 25: Ethereum Feature Importance

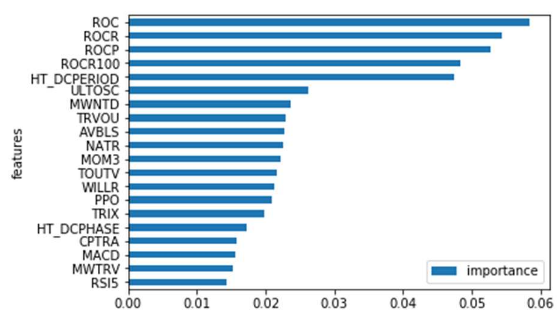

Figure 26: Litecoin Feature Importance

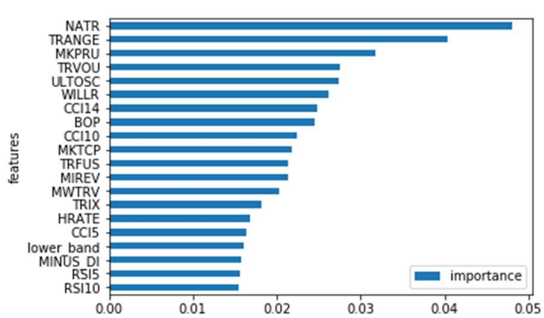

Figure 27: Ripple Feature Importance

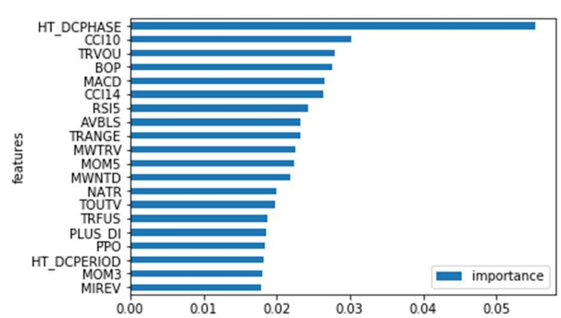

Figure 28: Zcash Feature Importance

Figure 23 to figure 28 clearly show that technical indicators, especially momentum indicators, are the essential part of the model prediction process. For Bitcoin, the top 3 important features are percentage price oscillator (PPO), normalized average true range (NATR) and 1-day rate of change (ROC) of a triple smooth EMA (TRIX). PPO is a technical momentum indicator that shows the relationship between a 26-period and a 12-period exponential moving average (EMA). It can help confirm trend direction and compare asset performance and volatility. NATR is normalized average true range and measures market volatility. TRIX is a momentum oscillator developed by Jack Hutson. It displays the percent rate of change of a triple EMA. It can generate signals to anticipate price reversals. In addition, from figure 23 we could see PPO is much important than other technical indicators and has an importance score of over 0.35. This suggests the model thinks price movements in the Bitcoin market are largely influenced by past momentums.

For EOS, the top 3 important features are Bitcoin average block size (AVBLS), Bitcoin exchange trade volume (TRVOU) and hash rate (HRATE). All of these are blockchain data. It implies the model would assign more weights on blockchain change than its historical price movements in predicting EOS returns. It also suggests EOS is easily influenced by cryptocurrencies market. For Ethereum, the top 3 important features are NATR, TRIX and balance of power (BOP), which is a technical momentum indicator that measures the market strength of buyers against sellers. It assesses the ability to drive market prices to the extreme from both sides. Like Bitcoin, the most important features in Ethereum, return predictions are all technical indicators even though they each have a relatively lower score.

This also applies to Litecoin, Ripple and Zcash, where technical momentum indicators are the most important part in the model prediction process. Rate of change (ROC), rate of change ratio (ROCR) and rate of change percentage (ROCP) are the three most important features in Litecoin return predictions. ROC measures the percentage change in price between the current price and the price n periods ago. For Ripple, the top three features are NATR, true range (TRANGE) and Bitcoin market price (MKPRU). This is consistent with our correlation analysis above that Ripple and Bitcoin are closely related cryptocurrency. Even the important technical indicators are the same in the model prediction process.

For Zcash, the top three features are dominant cycle phase using Hilbert transformation (HT_DCPHASE), 10-day commodity channel index (CCI10) and TRVOU. This means the model consider Zcash price movements having cyclical correlations so that dominant cycle phase is the most important feature. In addition, commodity channel Index is a momentum-based oscillator, which can identify cyclical trends, price reversals, price extremes and trend strength. These all suggest Zcash price movements display the characteristic of cyclical fluctuations.

6. Future Research Direction

Our analysis studies and compares different deep learning models in predicting cryptocurrency returns. We found LSTM networks consistently outperform feed-forward neural networks in bidirectional analysis and classification analysis. The structure of feed-forward neural networks does not allow them to store past information. Hence explains the better performance of LSTM networks. However, LSTM networks have a sharp drop in classification prediction. One reason we think accountable for this underperformance is the limited number of training data. In future research, we would like to try using intra-day data to address this problem. In addition, we would also like to explore the predictability of cryptocurrency by shortening the prediction time frame from one-day to a shorter-term. This is because, in a narrow time interval, the price is more driven by market power and momentum indicators than market news. Therefore, combining technical analysis with deep learning using intra-day data can be a future direction to further increase return prediction accuracy rates. s.

7. Conclusion

The combination of deep learning and technical analysis is effective in predicting cryptocurrency returns. The highest bidirectional accuracy rate we have is 76.11% for Bitcoin and the classification accuracy rate is 22.78%. For other cryptocurrencies, the average bidirectional accuracy rate is around 70% and the average classification accuracy rates is around 20%.

In both bidirectional and classification analysis, models based on the long short-term network structure outperform models based on feed-forward neural network. This outperformance of LSTM networks could be explained by its ability to store past information to help the model in predicting future returns. We then analyze the impact of different amount of past information on the LSTM model performance. After comparing different window length, we find the 2-layer LSTM model is best suitable for a window length shorter than 40 days while the Skeleton model is more fit for a window length longer than 100 days.

In classification analysis, we classify daily return into 21 non-overlap intervals and examine accuracy rates and the win ratio across models. We first compare accuracy rates using regression models to classification models and find regression models always beat classification models. We attribute the underperformance of classification models to the information loss when transforming numeric daily return into categorical data.

We then build up a trading strategy based on model predictions. Again, we find Sharpe ratios of trading strategies based on LSTM networks are higher than strategies using feed-forward networks. In addition, cumulative returns based on the Skeleton model predictions are higher or at least the same as the buy-and-hold benchmark for all cryptocurrencies. Finally, in order to understand the decision process behind the model, we conduct a feature importance analysis and found technical momentum indicators play the most important part in determining future returns. In future research, we would like to use intra-day data and shorten the prediction time frame to address the problem of limited sample size in cryptocurrency markets.

Appendix: Technical Indicator List

Acronym	Technical Indicator
BBAND WIDTH, UPPER SIGNAL, LOWER SIGNAL	Bollinger Bands
DEMA	Double Exponential Moving Average
EMA	Exponential Moving Average
H TRENDLINE	Hilbert Transform - Instantaneous
Trendline KMAM	Kaufman Adaptive Moving Average
MIDPOINT	MidPoint over period
MIDPRICE	Midpoint Price over period
SAR	Parabolic SAR
SAREXT	Parabolic SAR - Extended
SMA3, SMA5, SMA10, SMA20	Simple Moving Average
T	Triple Exponential Moving Average (T3)
TEMA	Triple Exponential Moving Average
TRIMA	Triangular Moving Average
WMA	Weighted Moving Average
ADX14, ADX20	Average Directional Movement Index
ADXR	Average Directional Movement Index Rating
APO	Absolute Price Oscillator
AROONOSC	Aroon Oscillator
BOP	Balance Of Power
CCI3, CCI5,CCI10,CCI14	Commodity Channel Index
CMO	Chande Momentum Oscillator
DX	Directional Movement Index
MACD, MACDSIGNAL, MACDHIST	Moving Average Convergence/Divergence
MINUS DI	Minus Directional Indicator
MINUS DM	Minus Directional Movement
MOM1, MOM3, MOM5, MOM10	Momentum
PLUS DI	Plus Directional Indicator
PLUS DM	Plus Directional Movement
PPO	Percentage Price Oscillator
ROC	Rate of change
ROCP	Rate of change Percentage
ROCR	Rate of change ratio
ROCR100	Rate of change ratio 100 scale
RSI5, RSI10, RSI14	Relative Strength Index
SLOWK, SLOWD	Stochastic
FASTK, FASTD	Stochastic Fast
TRIX	1-day Rate-Of-Change (ROC) of a Triple Smooth EMA
ULTOSC	Ultimate Oscillator
WILLR	Williams' %R
ATR	Average True Range
NATR	Normalized Average True Range
TRANGE	True Range
CDL2CROWS	Two Crows
CDL3BLACKCROWS	Three Black Crows
CDL3INSIDE	Three Inside Up/Down
CDL3LINESTRIKE	Three-Line Strike
CDL3OUTSIDE	Three Outside Up/Down
CDL3STARSINSOUTH	Three Stars In The South
CDL3WHITESOLDIERS	Three Advancing White Soldiers
CDLABANDONEDBABY	Abandoned Baby
CDLADVANCEBLOCK	Advance Block
CDLBELTHOLD	Belt-hold
CDLBREAKAWAY	Breakaway

Acronym	Technical Indicator
CDLCLOSINGMARUBOZU	Closing Marubozu
CDLCONCEALBABYSWALL	Concealing Baby Swallow
CDLCOUNTERATTACK	Counterattack
CDLDARKCLOUDCOVER	Dark Cloud Cover
CDLDOJI	Doji
CDLDOJISTAR	Doji Star
CDLDRAGONFLYDOJI	Dragonfly Doji
CDLENGULFING	Engulfing Pattern
CDLEVENINGDOJISTAR	Evening Doji Star
CDLEVENINGSTAR	Evening Star
CDLGAPSIDESIDEWHITE	Up/Down-gap side-by-side white lines
CDLGRAVESTONEDOJI	Gravestone Doji
CDLHAMMER	Hammer
CDLHANGINGMAN	Hanging Man
CDLHARAMI	Harami Pattern
CDLHARAMICROSS	Harami Cross Pattern
CDLHIGHWAVE	High-Wave Candle
CDLHIKKAKE	Hikkake Pattern
CDLHIKKAKEMOD	Modified Hikkake Pattern
CDLHOMINGPIGEON	Homing Pigeon
CDLIDENTICAL3CROWS	Identical Three Crows
CDLINNECK	In-Neck Pattern
CDLINVERTEDHAMMER	Inverted Hammer
CDLKICKING	Kicking
CDLKICKINGBYLENGTH	Kicking - bull/bear determined by the longer marubozu
CDLLADDERBOTTOM	Ladder Bottom
CDLLONGLEGGEDDOJI	Long Legged Doji
CDLLONGLINE	Long Line Candle
CDLMARUBOZU	Marubozu
CDLMATCHINGLOW	Matching Low
CDLMATHOLD	Mat Hold
CDLMORNINGDOJISTAR	Morning Doji Star
CDLMORNINGSTAR	Morning Star
CDLONNECK	On-Neck Pattern
CDLPIERCING	Piercing Pattern
CDLRICKSHAWMAN	Rickshaw Man
CDLRISEFALL3METHODS	Rising/Falling Three Methods
CDLSEPARATINGLINES	Separating Lines
CDLSHOOTINGSTAR	Shooting Star
CDLSHORTLINE	Short Line Candle
CDLSPINNINGTOP	Spinning Top
CDLSTALLEDPATTERN	Stalled Pattern
CDLSTICKSANDWICH	Stick Sandwich
CDLTAKURI	Takuri (Dragonfly Doji with very long lower shadow)
CDLTASUKIGAP	Tasuki Gap
CDLTHRUSTING	Thrusting Pattern
CDLTRISTAR	Tristar Pattern
BBAND WIDTH, UPPER SIGNAL, LOWER SIGNAL	Bollinger Bands
CDLUNIQUE3RIVER	Unique 3 River
CDLUPSIDEGAP2CROWS	Upside Gap Two Crows
CDLXSIDEGAP3METHODS	Upside/Downside Gap Three Methods
HT DCPERIOD	Hilbert Transform - Dominant Cycle Period
HT DCPHASE	Hilbert Transform - Dominant Cycle Phase
HT TRENDMODE	Hilbert Transform - Trend vs Cycle Mode

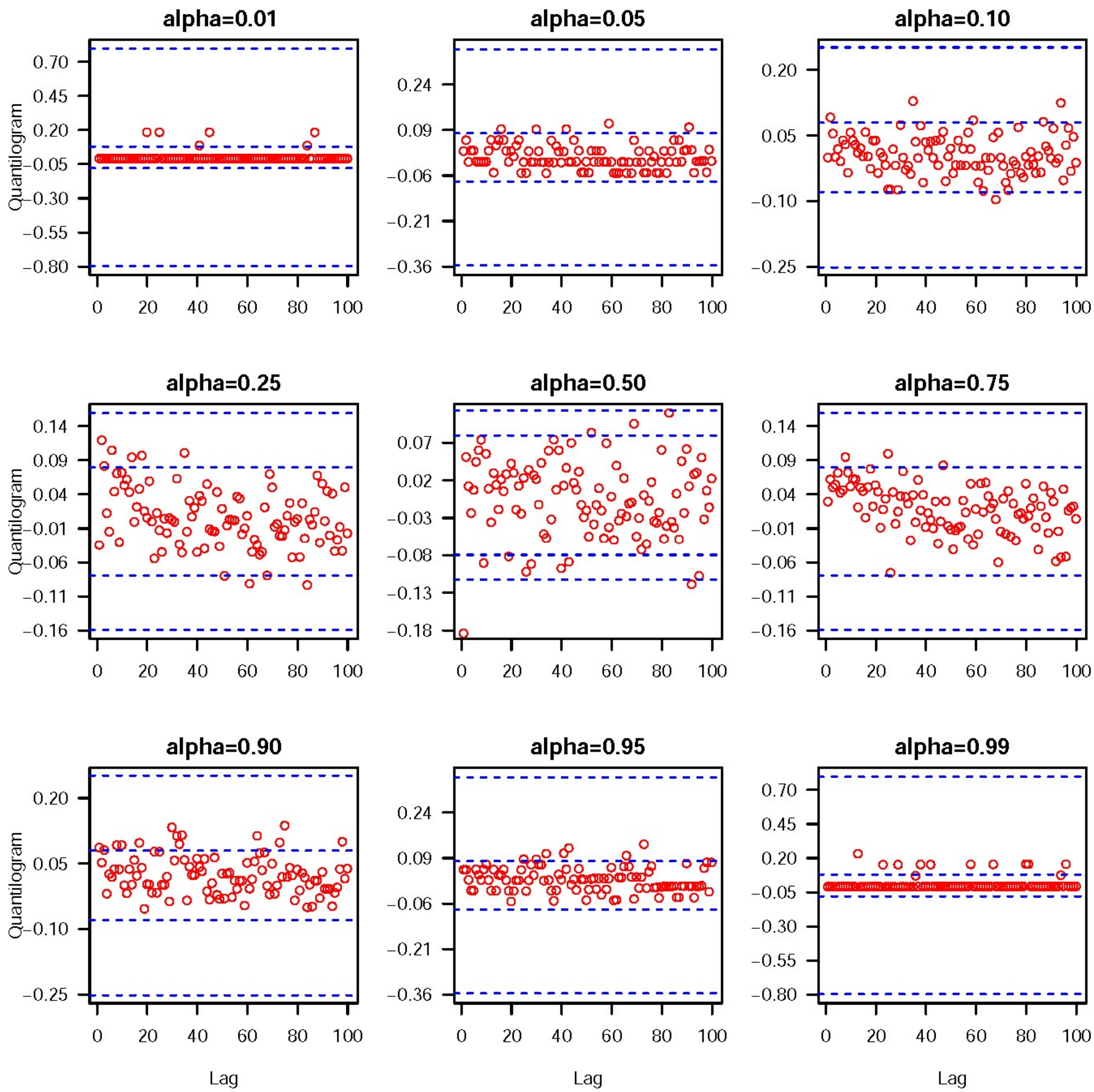

Figure A.1: Quantilogram test for EOS daily return

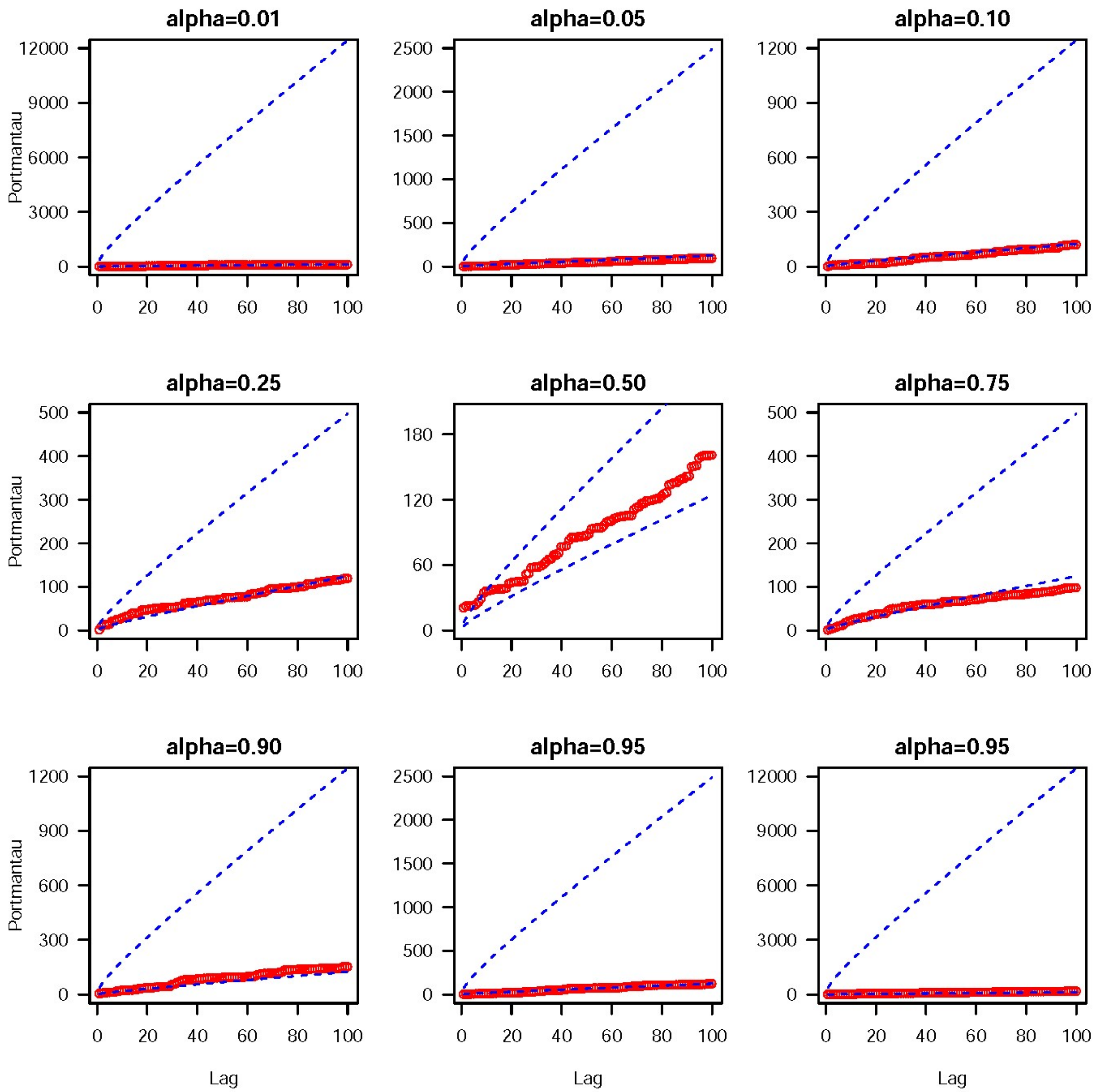

Figure A.2: Portmanteau test for EOS daily return

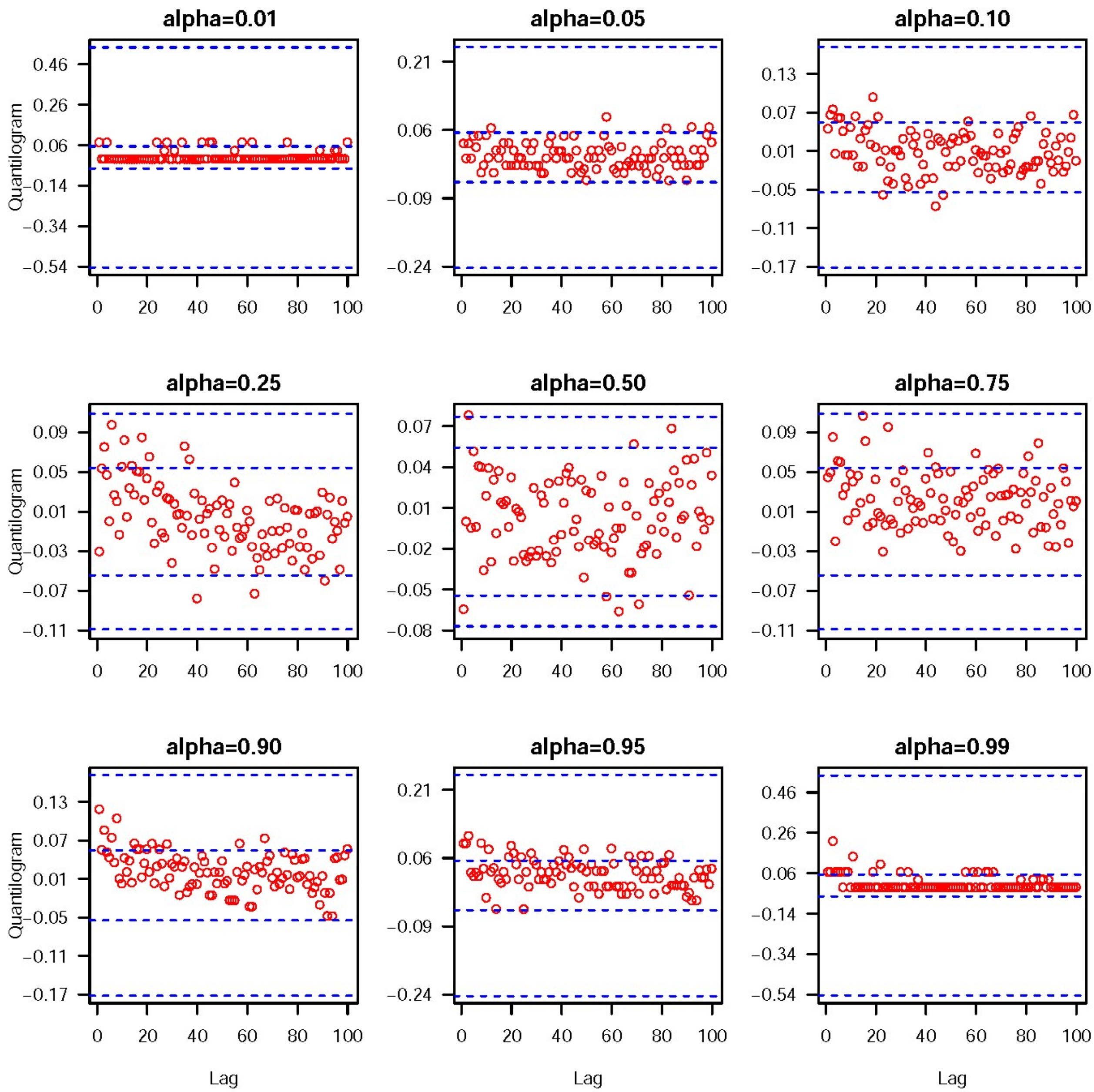

Figure A.3: Quantilogram test for Ethereum daily return

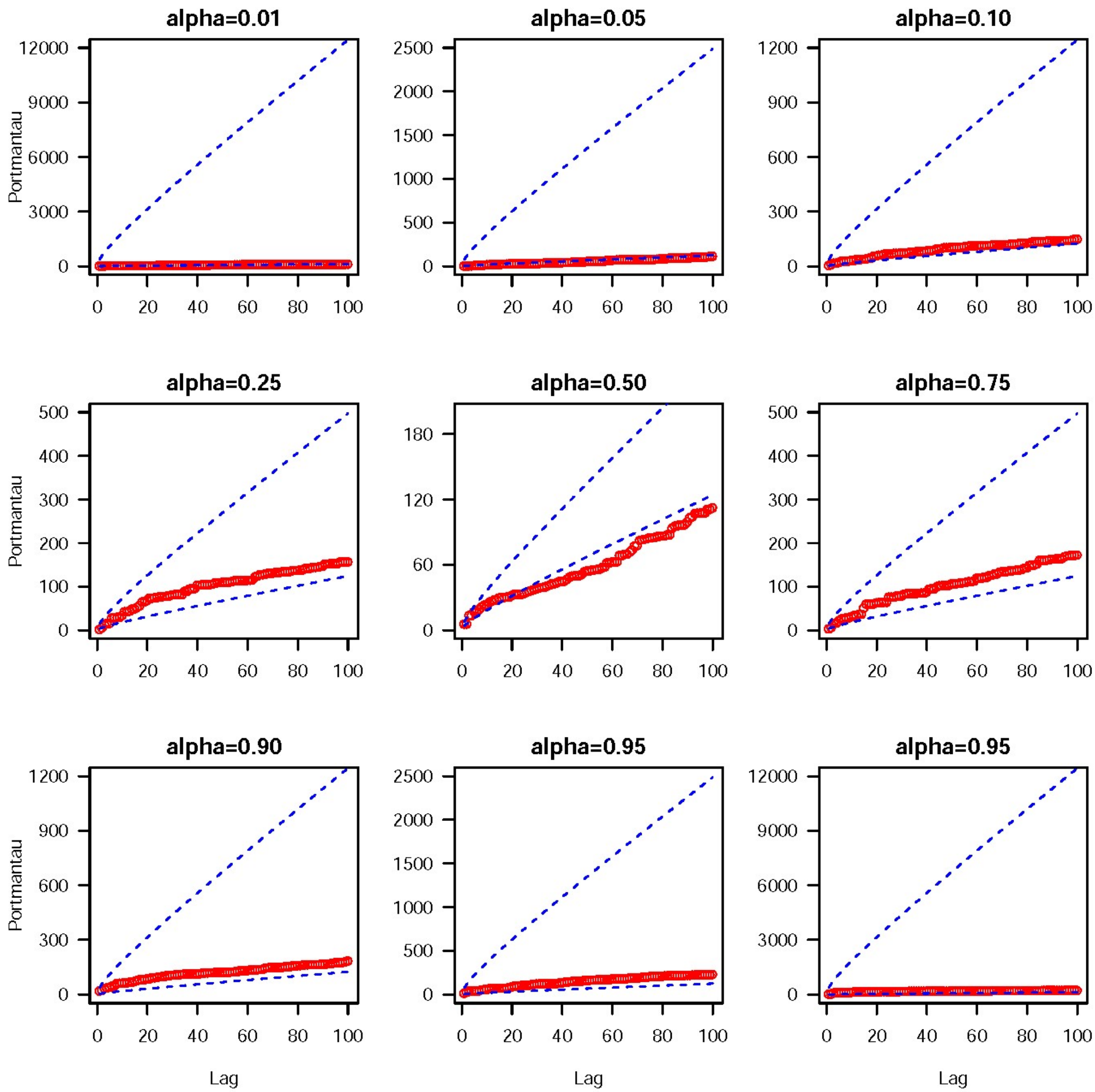

Figure A.4: Portmanteau test for Ethereum daily return

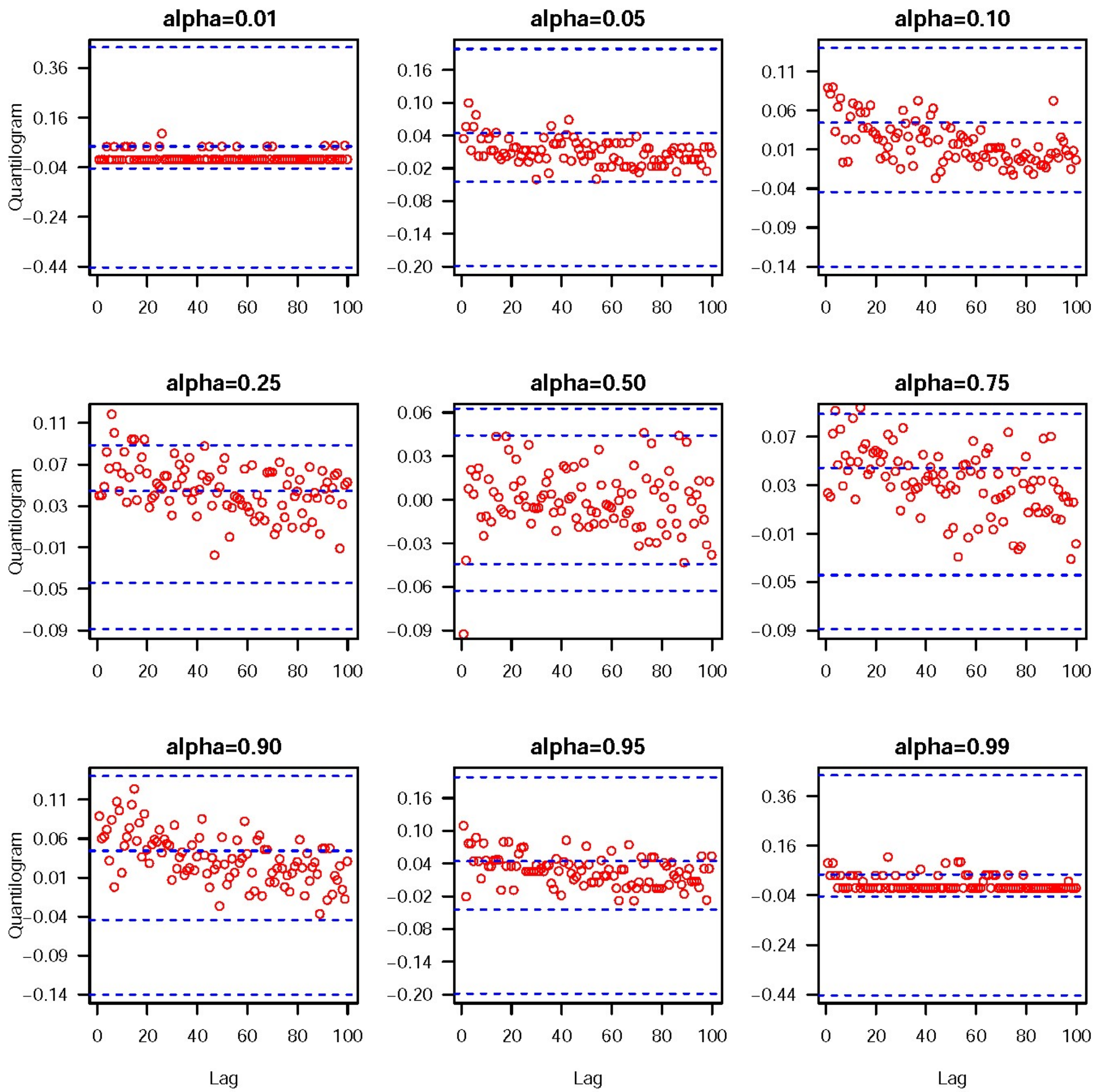

Figure A.5: Quantilogram test for Litecoin daily return

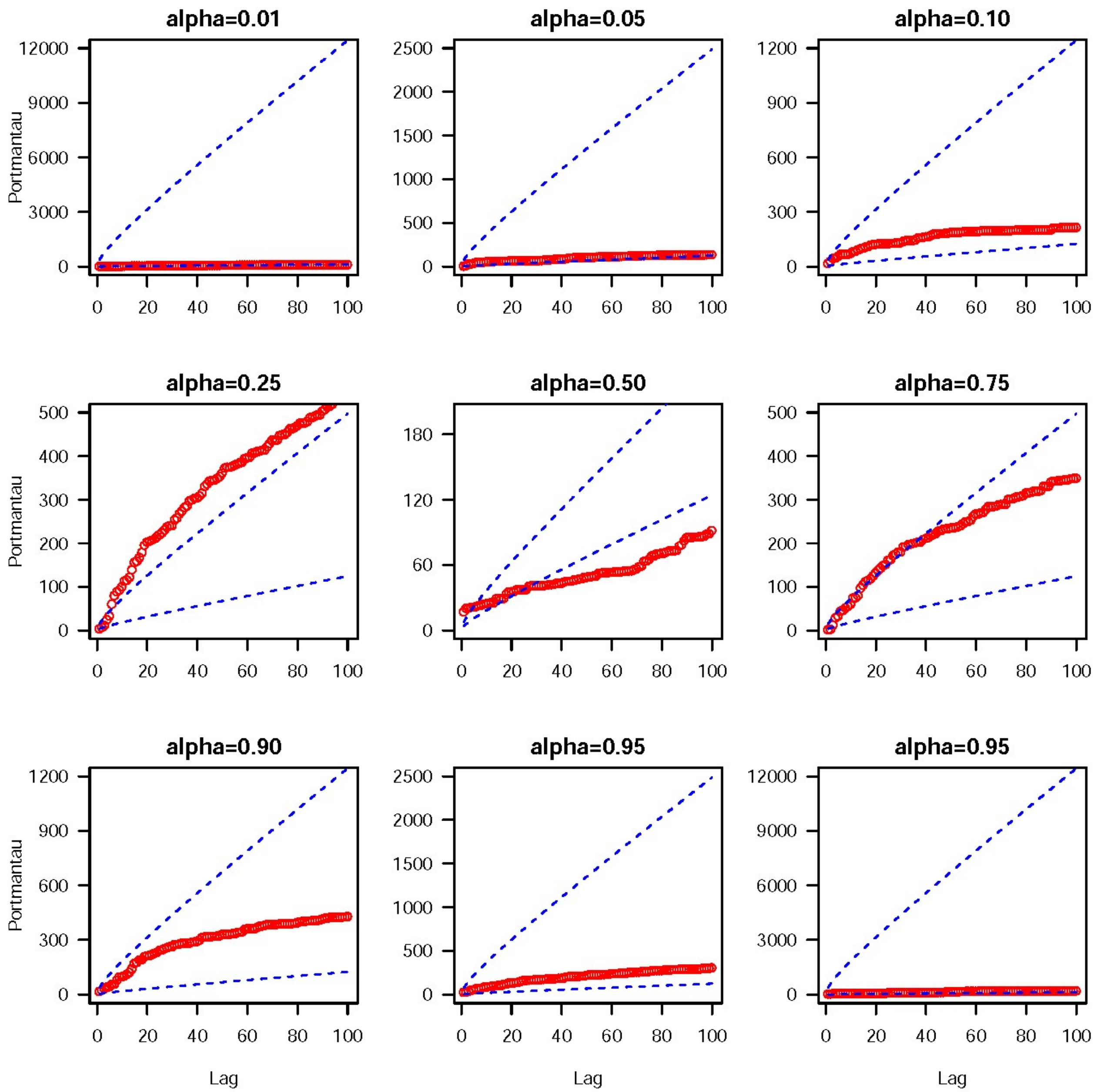

Figure A.6: Portmanteau test for Litecoin daily return

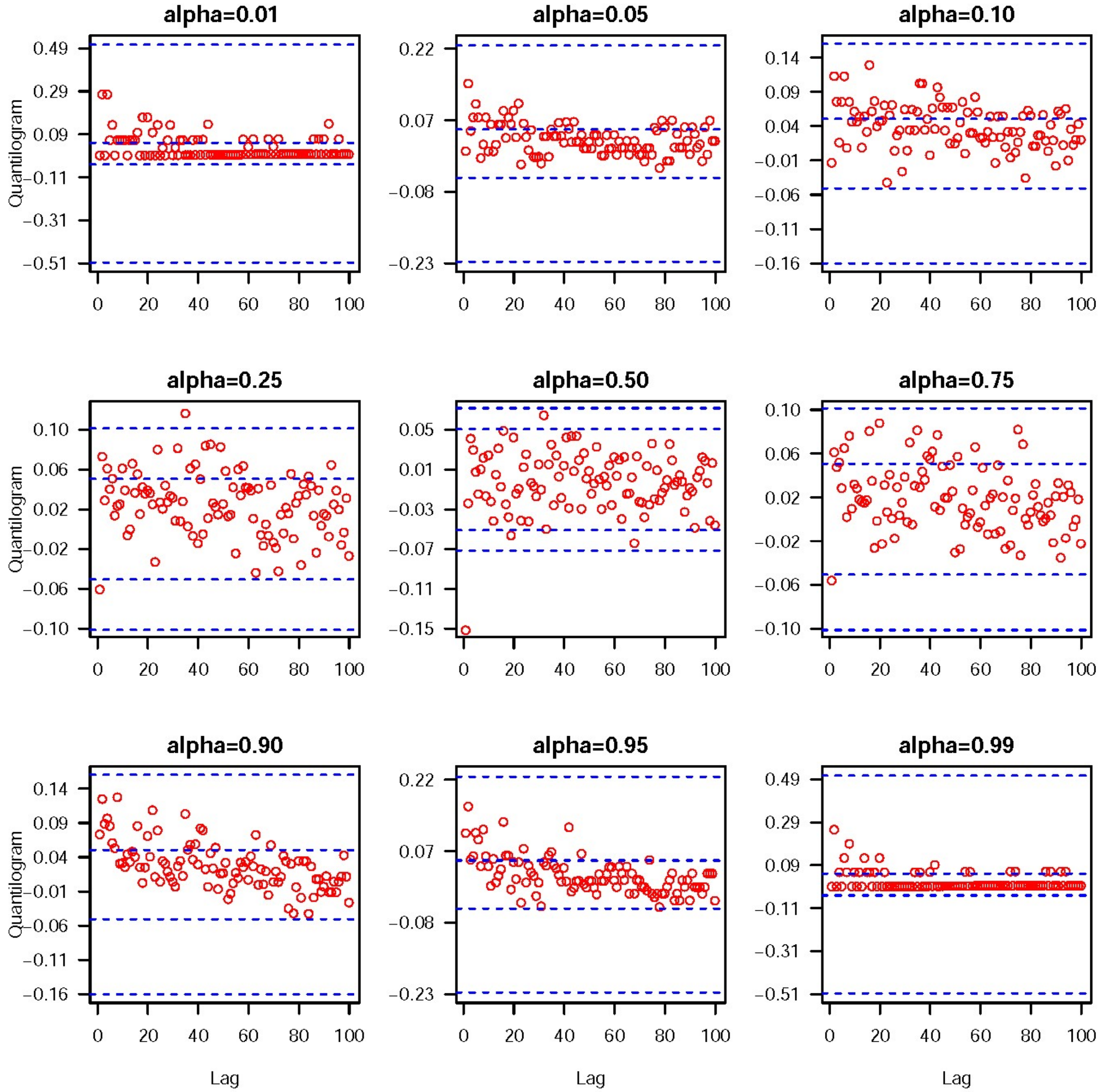

Figure A.7: Quantilogram test for Ripple daily return

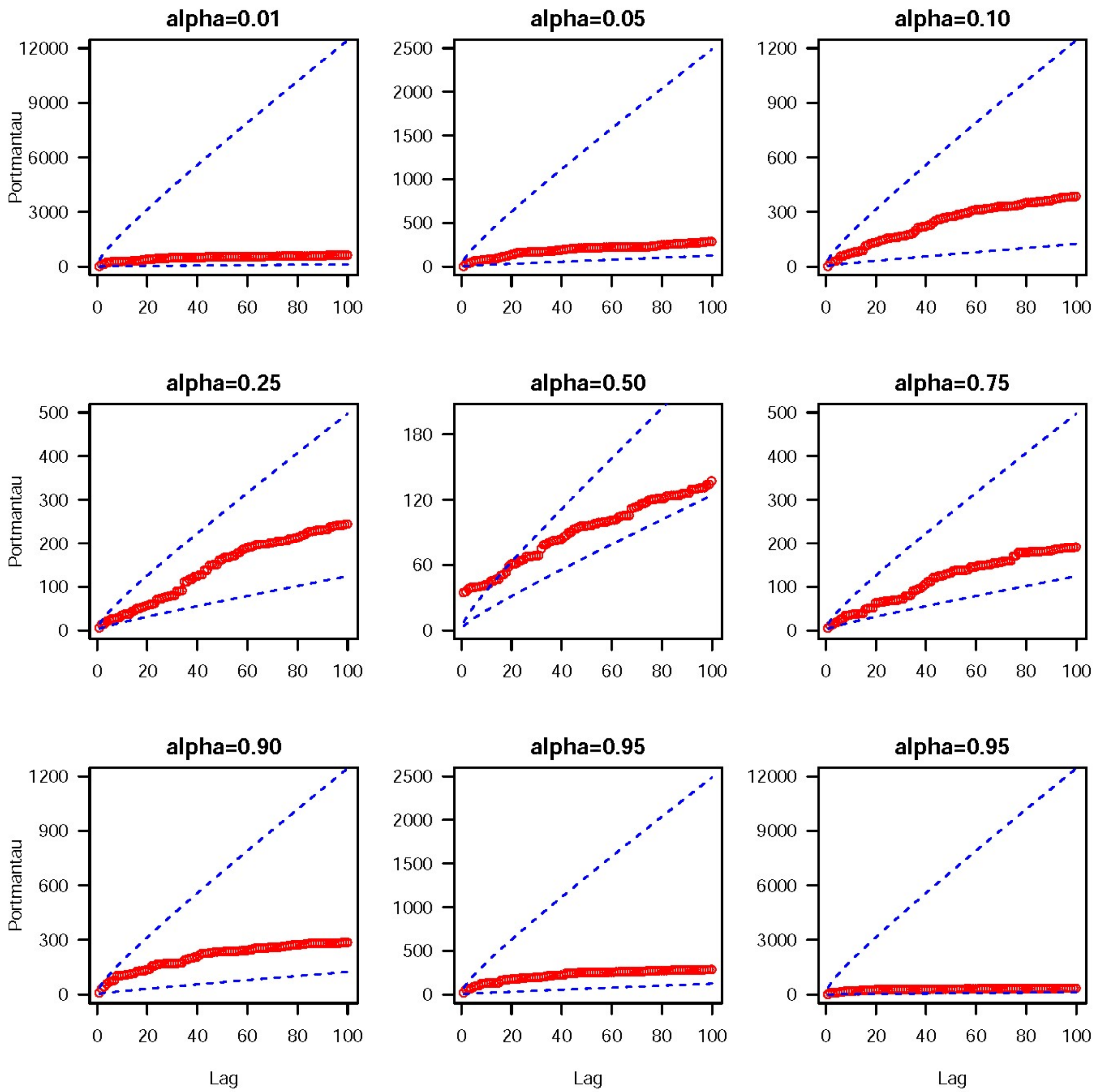

Figure A.8: Portmanteau test for Ripple daily return

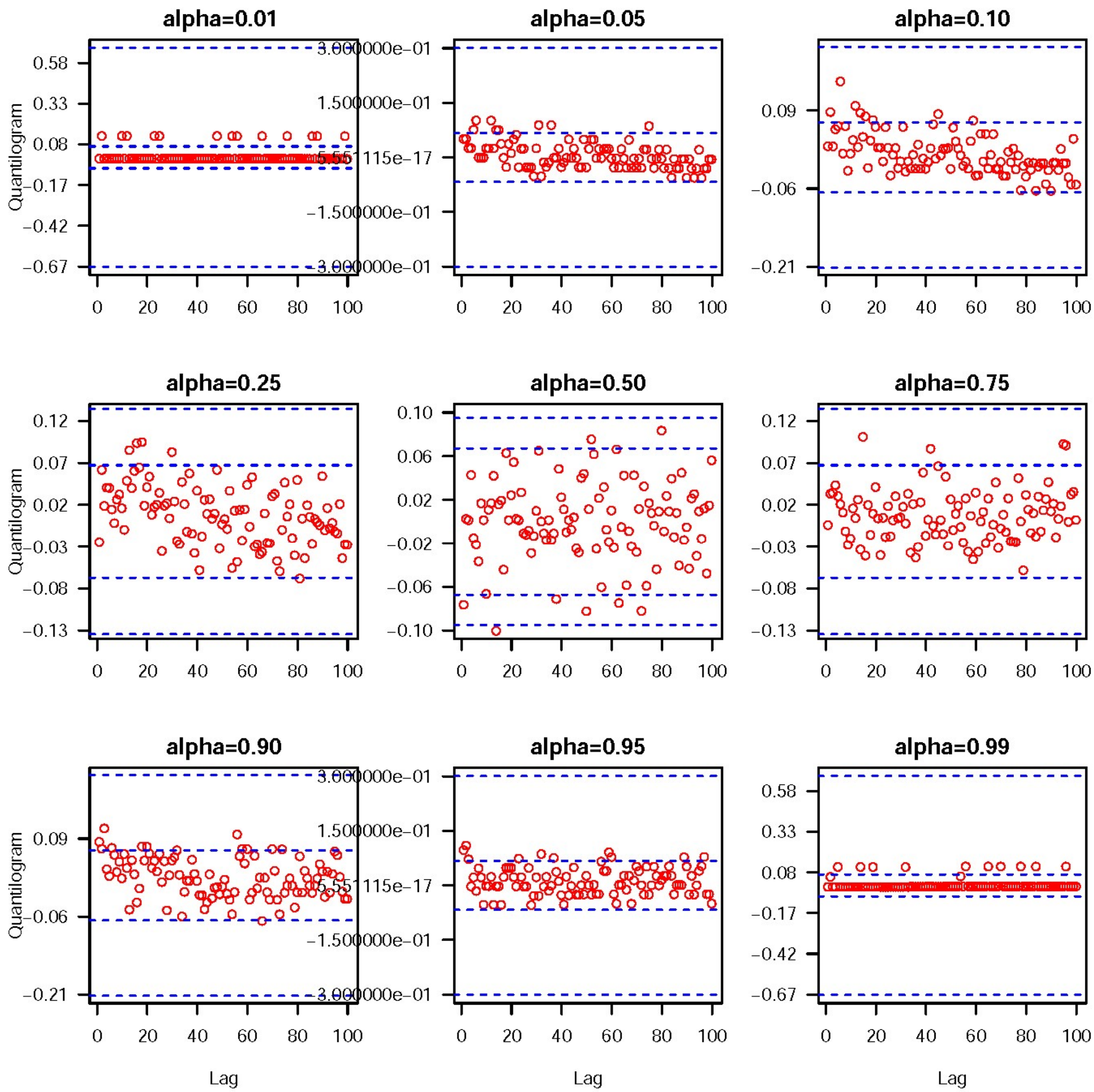

Figure A.9: Quantilogram test for Zcash daily return

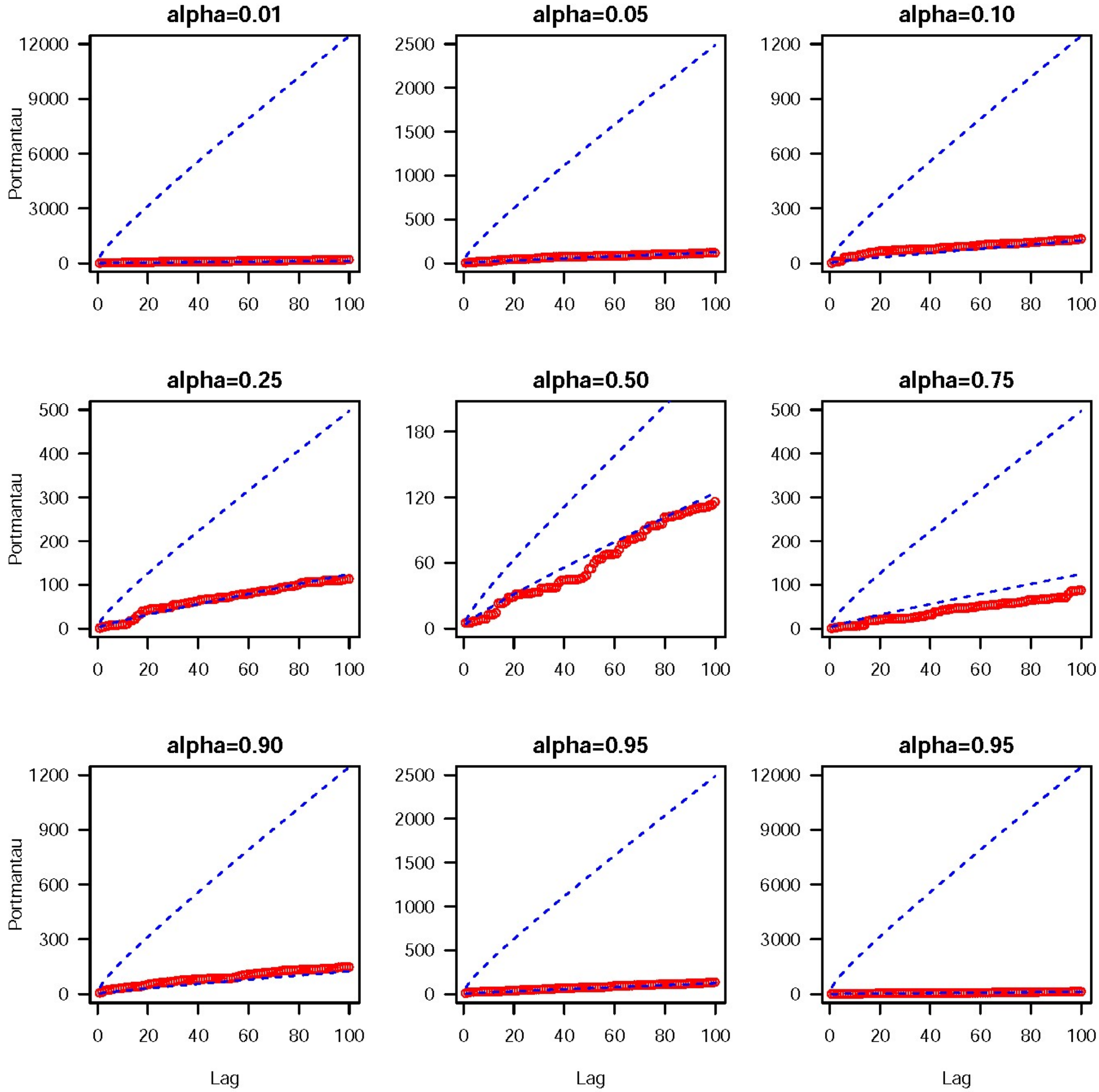

Figure A.10: Portmanteau test for Zcash daily return

Table-A.1: Win Ratio of the NN3 Model

Return Range	Bitcoin	Litecoin	Ripple	Zcash	Ethereum	EOS
-10	0.8333	0.5000	0.6098	0.7083	0.5000	0.4839
-9	-	-	0.6667	-	-	0.0000
-8	-	-	0.0000	0.5000	-	-
-7	-	1.0000	1.0000	0.0000	-	1.0000
-6	0.0000	-	1.0000	1.0000	1.0000	1.0000
-5	0.0000	-	0.0000	-	-	-
-4	1.0000	-	0.3333	0.6667	0.0000	0.0000
-3	1.0000	1.0000	-	1.0000	1.0000	-
-2	-	-	0.0000	0.0000	-	1.0000
-1	-	-	1.0000	0.0000	1.0000	0.5000
0	-	0.0000	0.3000	0.6000	0.0000	0.3333
1	1.0000	1.0000	1.0000	0.5000	-	0.6667
2	-	-	0.0000	1.0000	0.5000	0.5000
3	-	-	0.0000	0.6667	0.0000	0.5000
4	-	-	1.0000	0.5000	-	0.0000
5	-	-	0.3333	0.0000	0.6667	0.3333
6	1.0000	-	1.0000	-	-	-
7	-	0.0000	0.0000	0.0000	0.3333	-
8	-	0.0000	0.6667	0.5000	-	0.5000
9	-	-	0.7500	1.0000	0.5000	1.0000
10	0.3728	0.4767	0.4742	0.4706	0.4630	0.4643

Note: This table provides win ratio of the NN3 model range predictions. The 21 non-overlap range are indexed by number from -10 to 10. Win ratio is calculated by the number of true predictions over the frequency of the predicted range and "-" represents missing value because this range is not predicted by the NN3 model. The test sample period is from November 15th, 2018 to May 21st, 2019.

Table-A.2: Win Ratio of the Skeleton Model

Return Range	Bitcoin	Litecoin	Ripple	Zcash	Ethereum	EOS
-10	0.2857	-	-	-	-	-
-9	-	-	0.5000	-	-	-
-8	0.5000	-	0.5000	-	-	-
-7	0.5556	-	0.6154	-	-	-
-6	0.7500	-	0.3077	-	-	-
-5	0.0000	-	0.5500	-	-	-
-4	0.0000	-	0.4667	-	-	-
-3	-	-	0.5600	-	-	-
-2	0.3333	-	0.5000	-	0.5000	-
-1	0.4815	-	0.5714	-	0.4000	-
0	0.3906	-	0.4000	0.3000	0.5000	-
1	0.3659	-	0.0000	0.6667	0.6190	-
2	0.3333	-	-	0.2500	0.4118	0.0000
3	0.3333	-	-	0.6250	0.5000	-
4	-	-	-	0.4286	0.4167	0.0000
5	-	-	-	0.5714	0.5714	-
6	-	-	-	0.7500	0.5000	-
7	-	-	-	0.4286	0.0833	-
8	-	-	-	0.8333	0.0000	1.0000
9	-	-	-	0.4545	0.2500	1.0000
10	-	0.4778	-	0.5221	0.6000	0.4886

Note: This table provides win ratio of the Skeleton model range predictions. The 21 non-overlap range are indexed by number from -10 to 10. Win ratio is calculated by the number of true predictions over the frequency of the predicted range and "-" represents missing value because this range is not predicted by the Skeleton model. The test sample period is from November 15th, 2018 to May 21st, 2019.

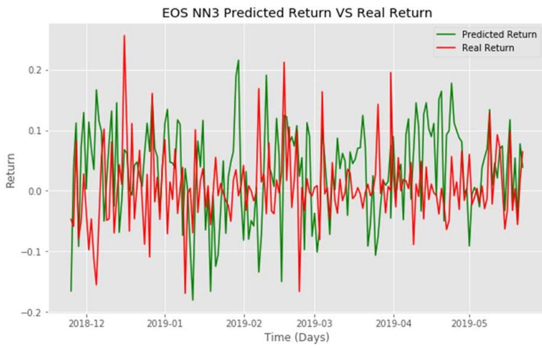

Figure A.11: EOS NN3 Predicted Return

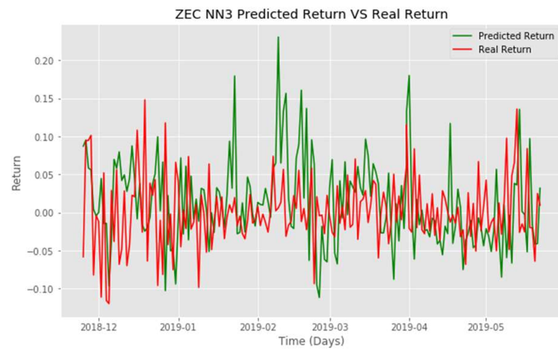

Figure A.12: Zcash NN3 Predicted Return

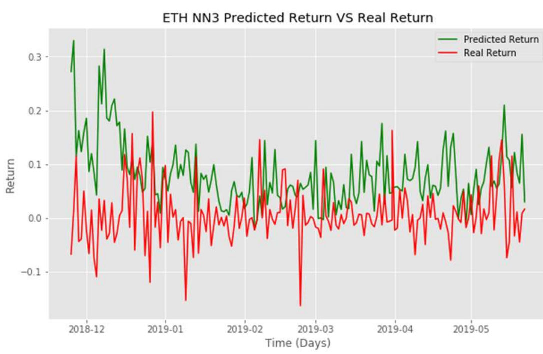

Figure A.13: Ethereum NN3 Predicted Return

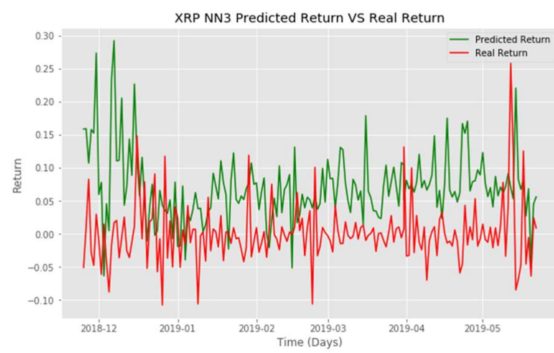

Figure A.14: Ripple NN3 Predicted Return

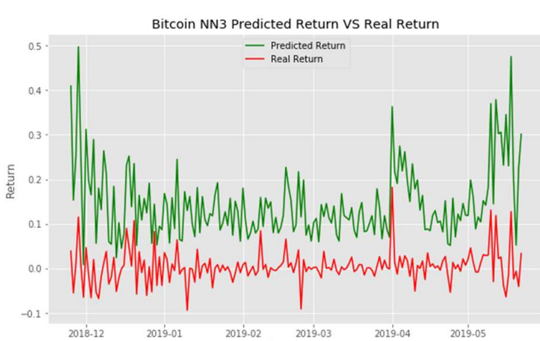

Figure A.15: Bitcoin NN3 Predicted Return

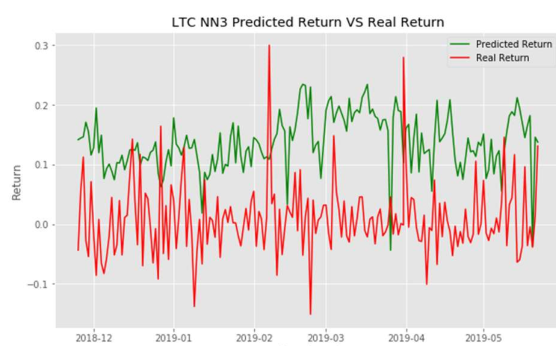

Figure A.16: Litecoin NN3 Predicted Return

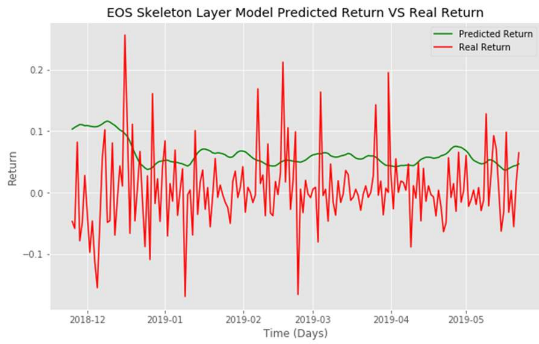

Figure A.17: EOS Skeleton Predicted Return

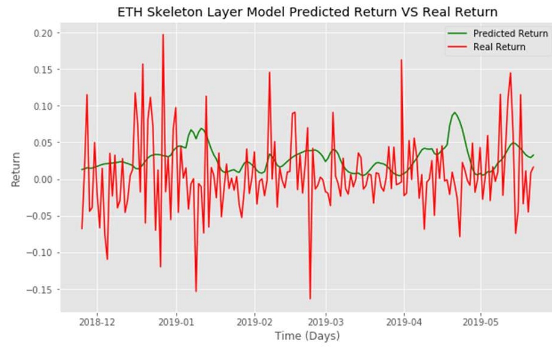

Figure A.18: ETH Skeleton Predicted Return

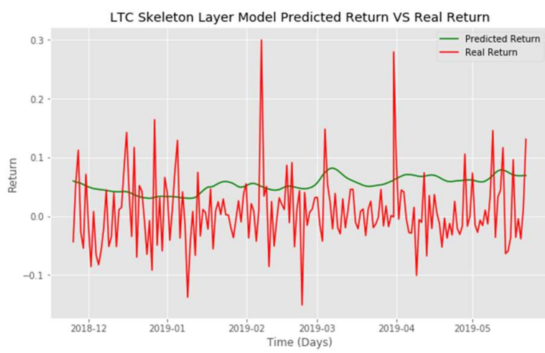

Figure A.19: LTC Skeleton Predicted Return

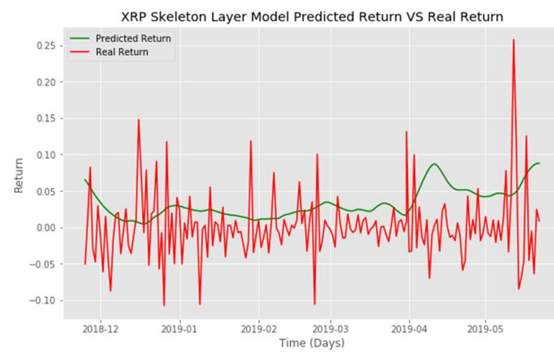

Figure A.20: XRP Skeleton Predicted Return

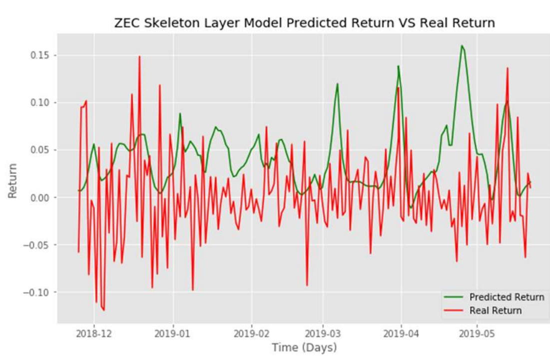

Figure A.21: ZEC Skeleton Predicted Return

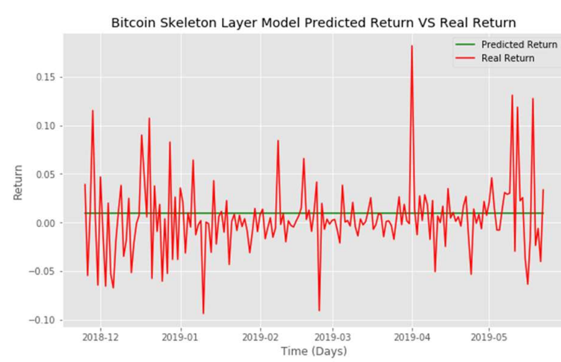

Figure A.22: Bitcoin Skeleton Predicted Return

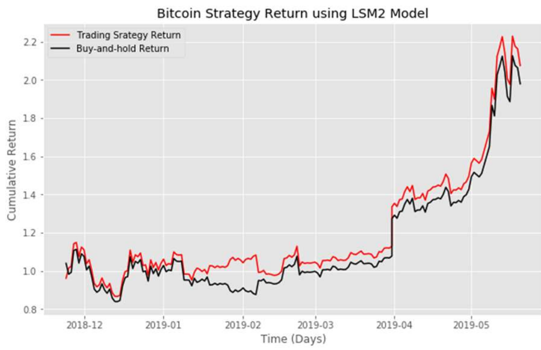

Figure A.23: LSTM2 Cumulative Return for Bitcoin

Figure A.24: LSTM2 Cumulative Return for Litecoin

Figure A.25: LSTM2 Cumulative Return for Ripple

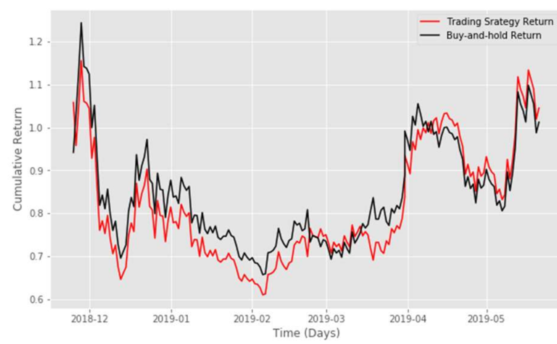

Figure A.26: LSTM2 Cumulative Return for Zcash

Figure A.27: LSTM2 Cumulative Return for EOS

Figure A.28: LSTM2 Cumulative Return for Ethereum

Figure A.29: NN3 Cumulative Return for EOS

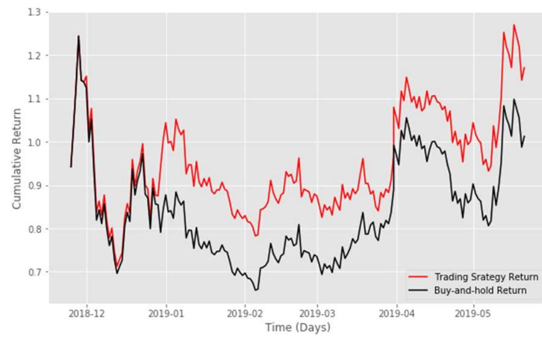

Figure A.30: NN3 Cumulative Return for Zcash

Figure A.31: NN3 Cumulative Return for Ethereum

Figure A.32: NN3 Cumulative Return for Ripple

Figure A.33: NN3 Cumulative Return for Litecoin

Figure A.34: NN3 Cumulative Return for Bitcoin

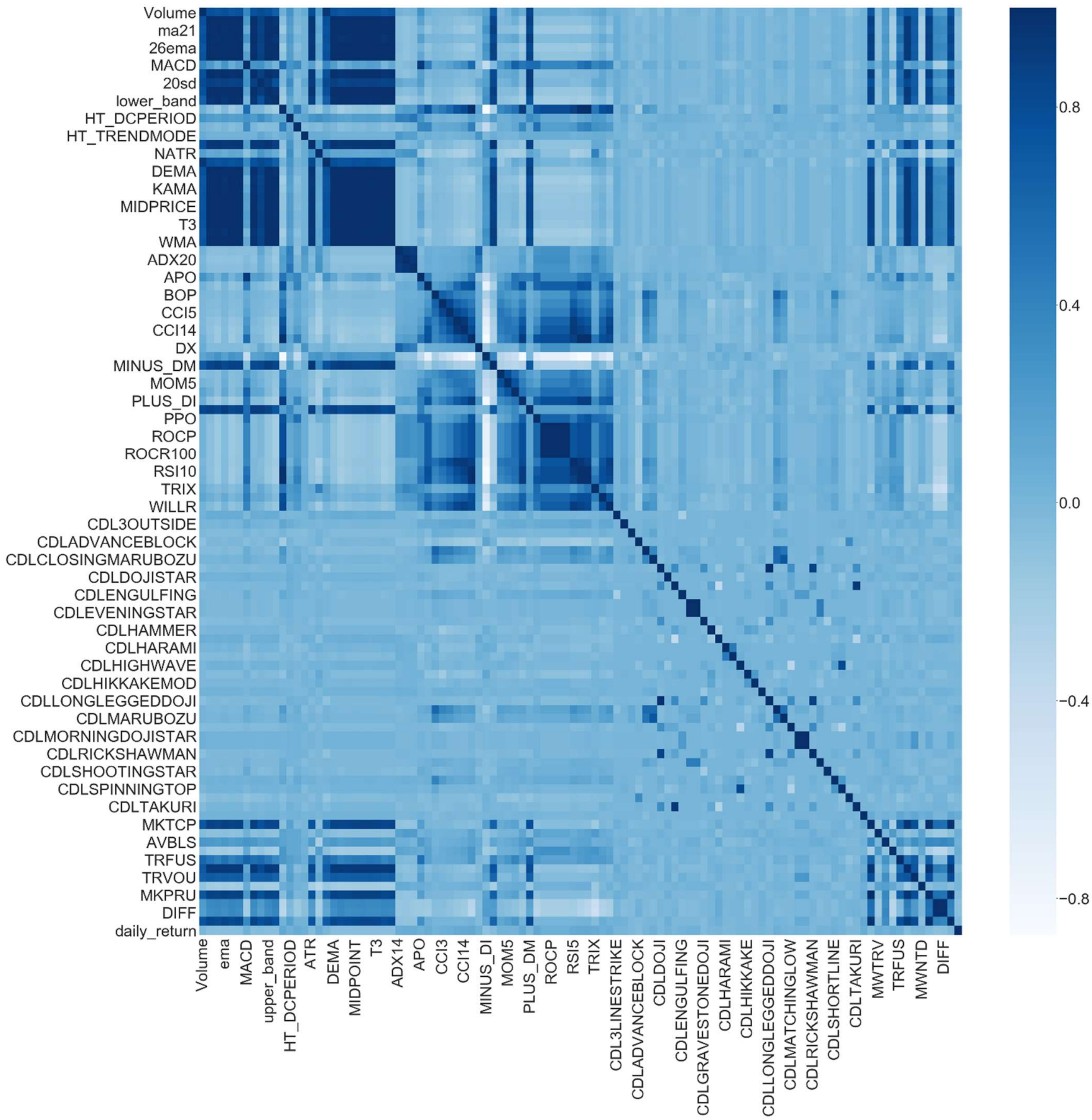

Figure A.35: EOS Correlation Analysis between Features and the Target

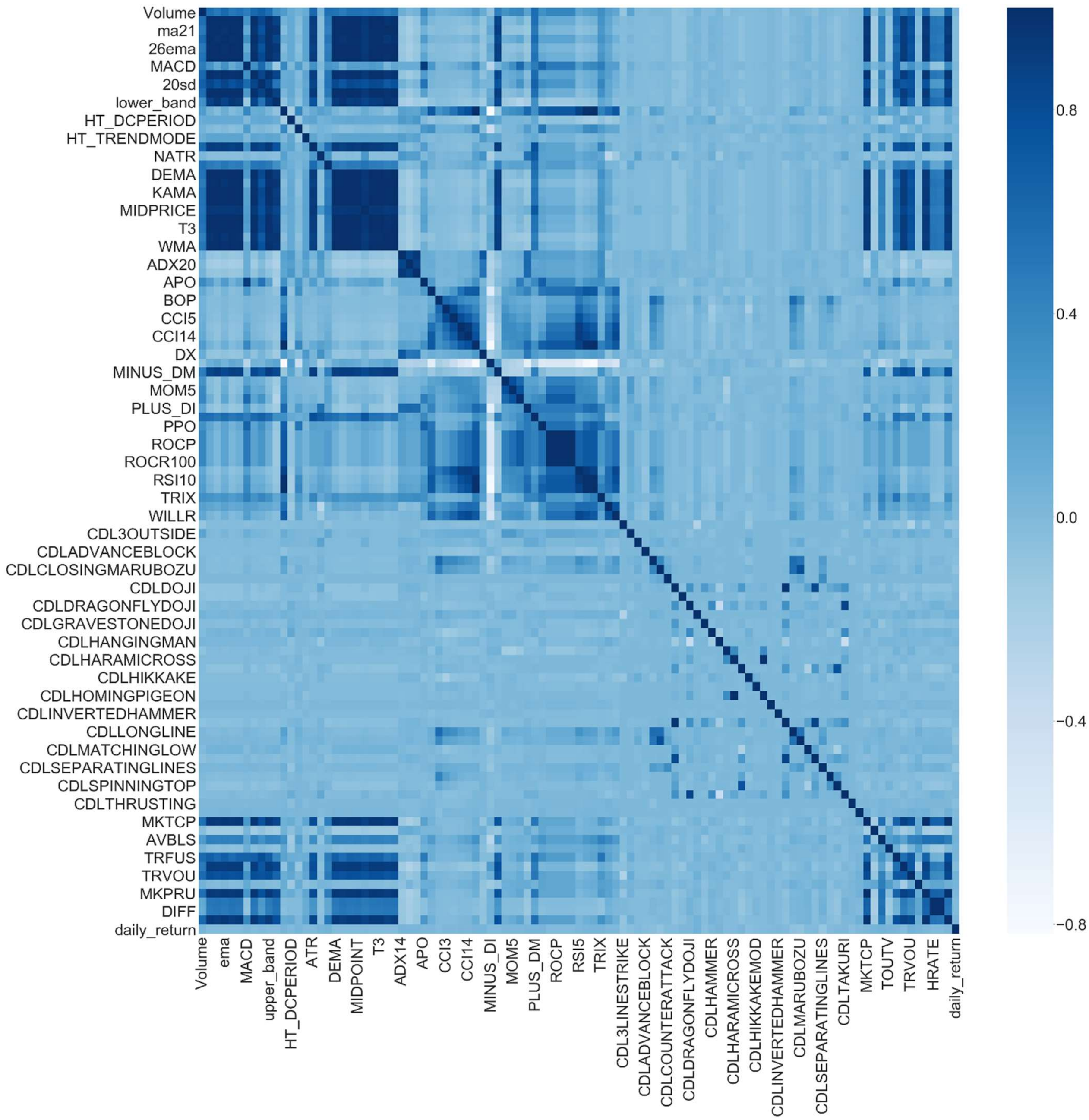

Figure A.36: Ethereum Correlation Analysis between Features and the Target

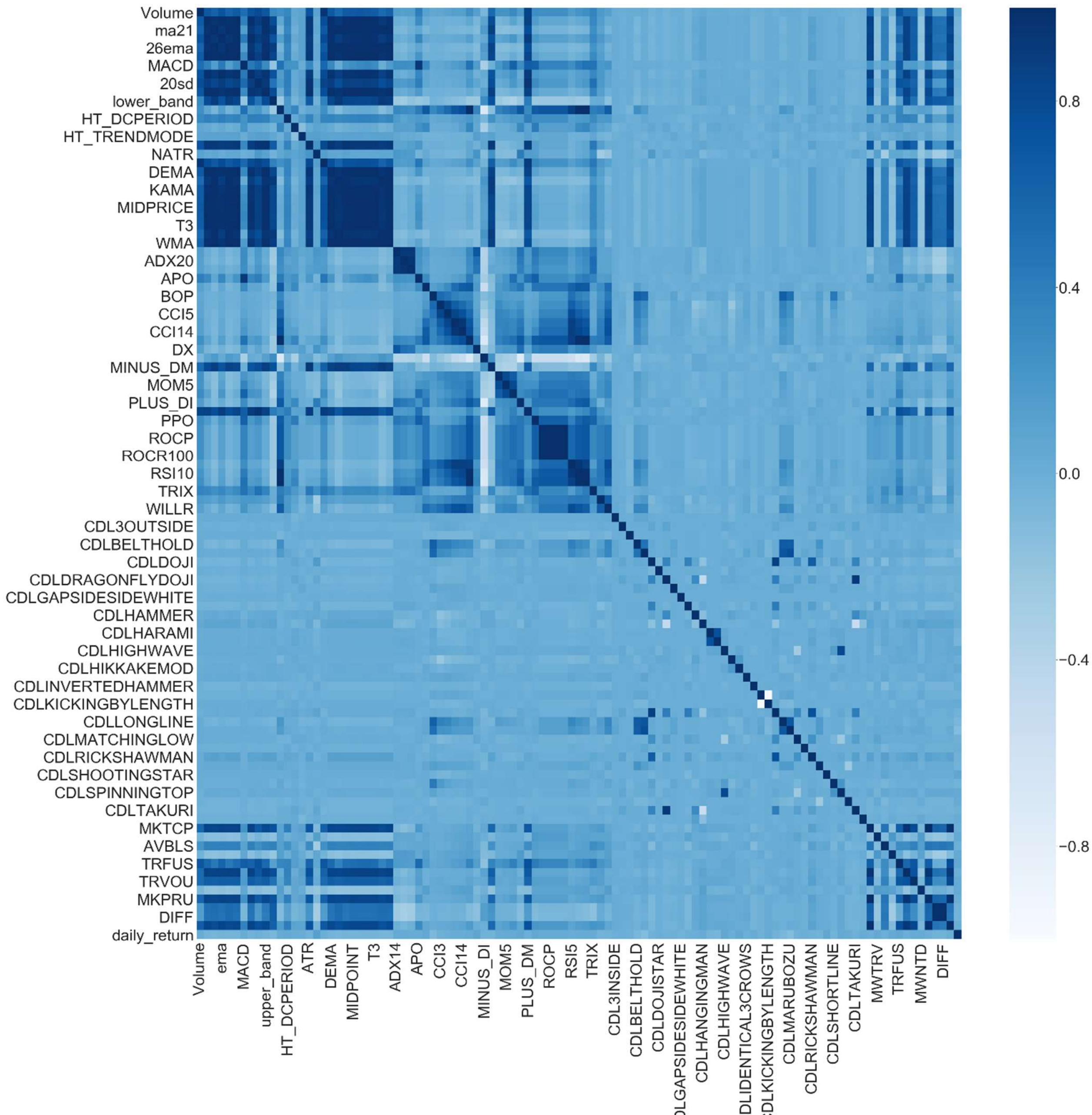

Figure A.37: Litecoin Correlation Analysis between Features and the Target

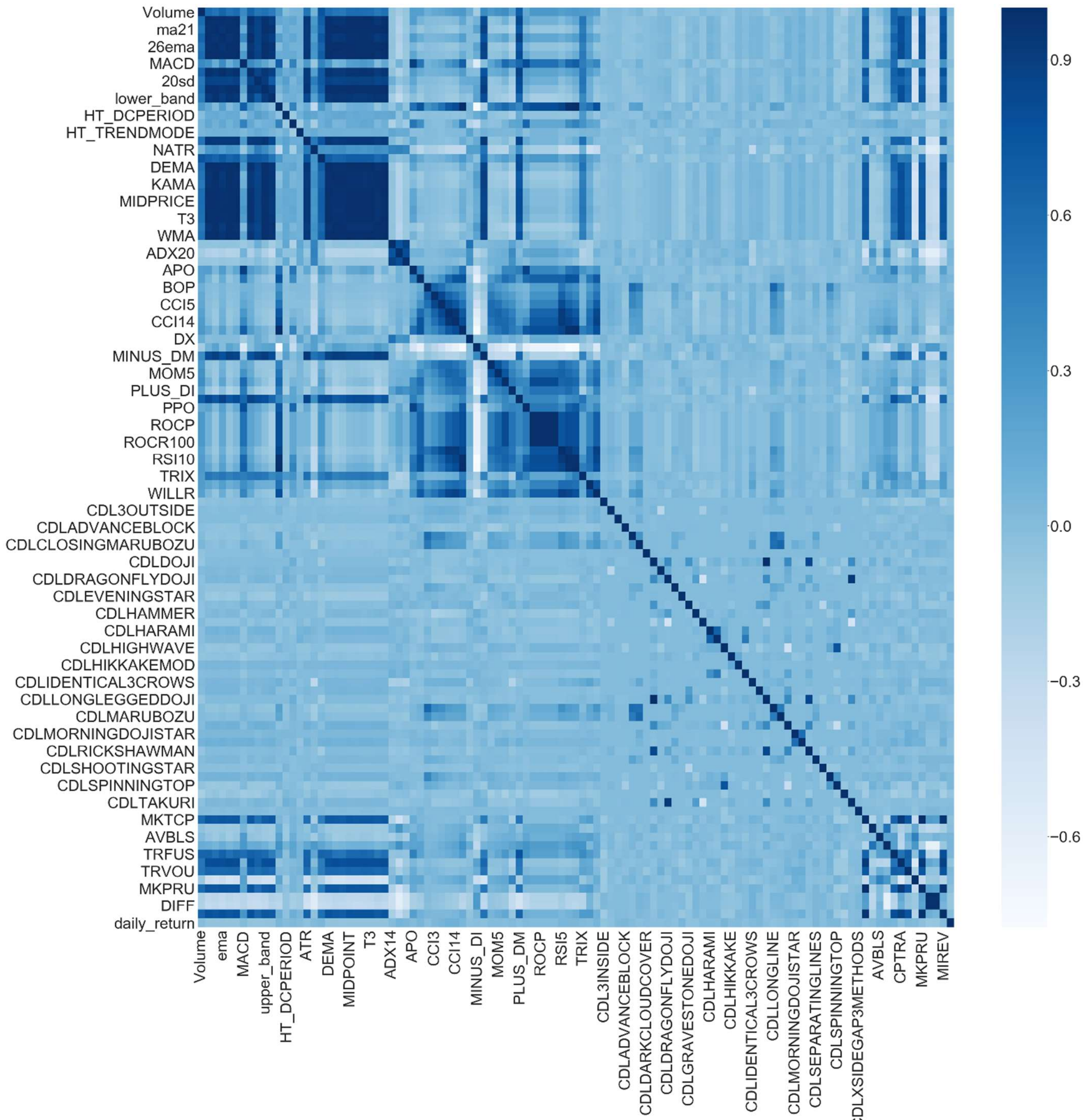

Figure A.38: Ripple Correlation Analysis between Features and the Target

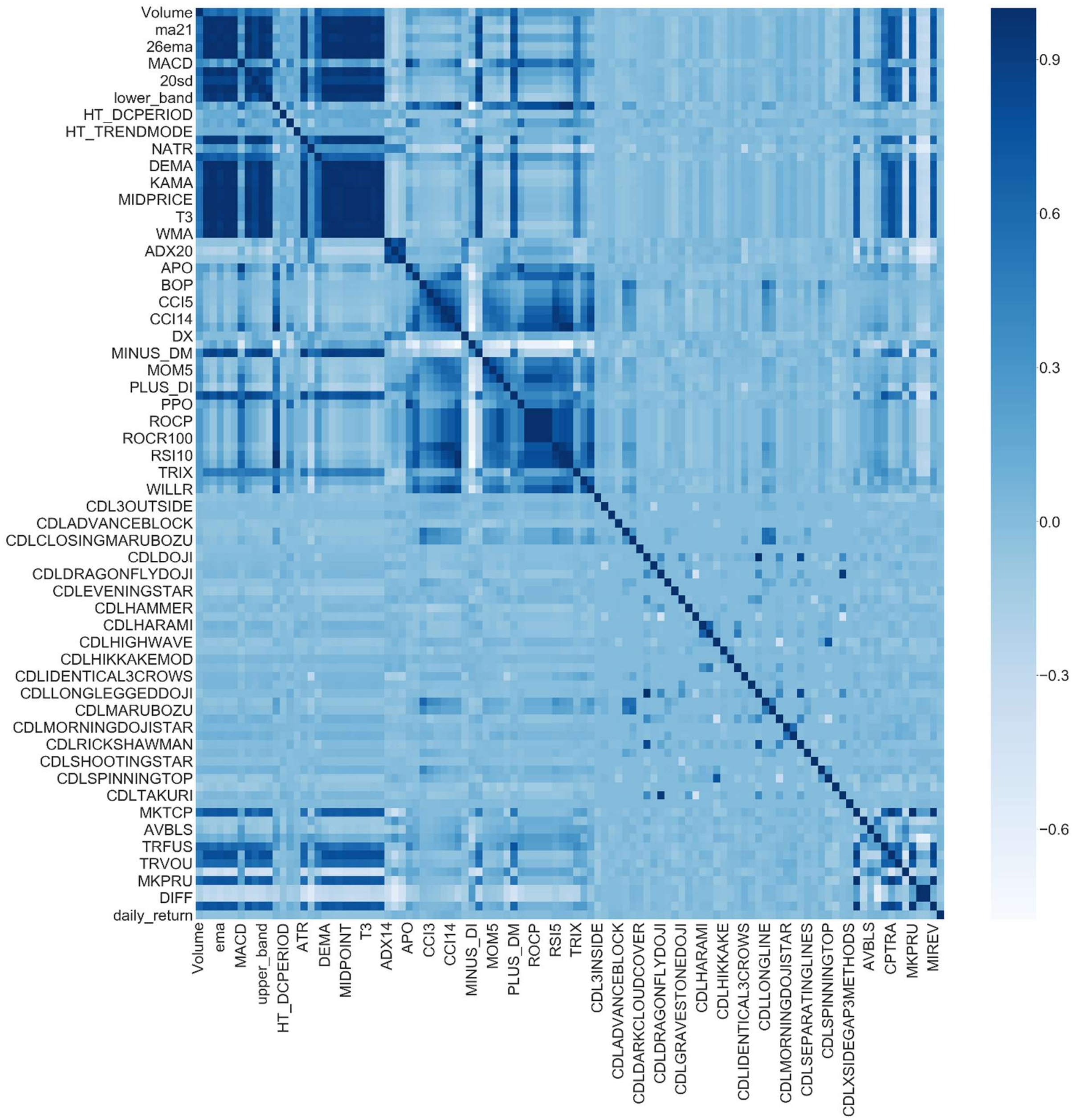

Figure A.38: Zcash Correlation Analysis between Features and the Target

References

[1] Huang, Jing-Zhi, William Huang, and Jun Ni. "Predicting Bitcoin Returns Using High-Dimensional Technical Indicators." *The Journal of Finance and Data Science* (2018).

[2] McNally, Sean, Jason Roche, and Simon Caton. "Predicting the price of Bitcoin using Machine Learning." In 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), pp. 339-343. IEEE, 2018.

[3] Phillips, Peter CB, Shuping Shi, and Jun Yu. "Testing for multiple bubbles: Historical episodes of exuberance and collapse in the S&P 500." *International economic review* 56, no. 4 (2015): 1043-1078.

[4] Alessandretti, Laura, Abeer ElBahrawy, Luca Maria Aiello, and Andrea Baronchelli. "Anticipating cryptocurrency prices using machine learning." *Complexity* 2018 (2018).

[5] Colianni, Stuart, Stephanie Rosales, and Michael Signorotti. "Algorithmic trading of cryptocurrency based on Twitter sentiment analysis." *CS229 Project* (2015): 1-5.

[6] Detzel, Andrew, Hong Liu, Jack Strauss, Goufu Zhou, and Yingzi Zhu. "Bitcoin: Predictability and profitability via technical analysis." *SSRN Electron. J.* (2018).

[7] Feng, Guanhao, Jingyu He, and Nicholas G. Polson. "Deep learning for predicting asset returns." *arXiv preprint arXiv:1804.09314* (2018).

[8] Fischer, Thomas Günter, Christopher Krauss, and Alexander Deinert. "Statistical Arbitrage in Cryptocurrency Markets." *Journal of Risk and Financial Management* 12, no. 1 (2019): 31.

[9] Geuder, Julian, Harald Kinateder, and Niklas F. Wagner. "Cryptocurrencies as financial bubbles: The case of Bitcoin." *Finance Research Letters* (2018).

[10] Gu, Shihao, Bryan Kelly, and Dacheng Xiu. *Empirical asset pricing via machine learning*. No. w25398. National Bureau of Economic Research, 2018.

[11] Jang, Huisu, and Jaewook Lee. "An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information." *Ieee Access* 6 (2017): 5427-5437.

[12] Liu, Yukun, and Aleh Tsyvinski. *Risks and returns of cryptocurrency*. No. w24877. National Bureau of Economic Research, 2018.

[13] Phillips, Ross C., and Denise Gorse. "Predicting cryptocurrency price bubbles using social media data and epidemic modelling." In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1-7. IEEE, 2017.

[14] Shah, Devavrat, and Kang Zhang. "Bayesian regression and Bitcoin." In *2014 52nd annual Allerton conference on communication, control, and computing (Allerton)*, pp. 409-414. IEEE, 2014.

[15] Vo, Nhi NY, and Guandong Xu. "The volatility of Bitcoin returns and its correlation to financial markets." In *2017 International Conference on Behavioral, Economic, Socio-cultural Computing (BESC)*, pp. 1-6. IEEE, 2017.

[16] Zhu, Wentao, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks." In *Thirtieth AAAI Conference on Artificial Intelligence*. 2016.

[17] Lahmiri, Salim, and Stelios Bekiros. "Cryptocurrency forecasting with deep learning chaotic neural networks." *Chaos, Solitons & Fractals* 118 (2019): 35-40.

[18] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980*(2014).

[19] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9, no. 8 (1997): 1735-1780.

[20] Hochreiter, Sepp, and Jürgen Schmidhuber. "LSTM can solve hard long time lag problems." In *Advances in neural information processing systems*, pp. 473-479. 1997.

[21] Hochreiter, Sepp, and Jürgen Schmidhuber. "Flat minima." *Neural Computation* 9, no. 1 (1997): 1-42.

[22] Wilder, J. Welles. *New concepts in technical trading systems*. Trend Research, 1978.

[23] Wilder, David A. "Predictability of behaviors, goals, and unit of perception." *Personality and Social Psychology Bulletin* 4, no. 4 (1978): 604-607.

[24] Li, Tianyu Ray, Anup Chamrajnagar, Xander Fong, Nicholas Rizik, and Feng Fu. "Sentiment-based prediction of alternative cryptocurrency price fluctuations using gradient boosting tree model." *Frontiers in Physics* 7 (2019): 98.

[25] Lamon, Connor, Eric Nielsen, and Eric Redondo. "Cryptocurrency price prediction using news and social media sentiment." *SMU Data Sci. Rev* 1, no. 3 (2017): 1-22.

[26] Abraham, Jethin, Daniel Higdon, John Nelson, and Juan Ibarra. "Cryptocurrency price prediction using tweet volumes and sentiment analysis." *SMU Data Science Review* 1, no. 3 (2018): 1.

[27] Brown, David P., and Robert H. Jennings. "On technical analysis." *The Review of Financial Studies* 2, no. 4 (1989): 527-551.

[28] Lo, Andrew W., Harry Mamaysky, and Jiang Wang. "Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation." *The journal of finance* 55, no. 4 (2000): 1705-1765.

[29] Harvey, Campbell R. "Bitcoin myths and facts." *Available at SSRN 2479670* (2014).

[30] Bollinger, John. "Using bollinger bands." *Stocks & Commodities* 10, no. 2 (1992): 47-51.

[31] Bollinger, John J., Wayne M. Itano, David J. Wineland, and D. J. Heinzen. "Optimal frequency measurements with maximally correlated states." *Physical Review A* 54, no. 6 (1996): R4649.

[32] Blume, Lawrence, David Easley, and Maureen O'hara. "Market statistics and technical analysis: The role of volume." *The Journal of Finance* 49, no. 1 (1994): 153-181.

[33] Bessembinder, Hendrik, and Kalok Chan. "Market efficiency and the returns to technical analysis." *Financial management*(1998): 5-17.

[34] Lambert, Donald R. "Commodity channel index: Tool for trading cyclic trends." *Technical Analysis of Stocks & Commodities* 1 (1983): 47.

[35] Linton, Oliver B., and Yoon-Jae Whang. "A quantilogram approach to evaluating directional predictability." (2004).

[36] Han, H., Linton, O., Oka, T., & Whang, Y. J. (2016). The cross-quantilogram: Measuring quantile dependence and testing directional predictability between time series. *Journal of Econometrics*, *193*(1), 251-270.

[37] Breiman, Leo. *Classification and regression trees*. Routledge, 2017.

[38] Fischer, Thomas, and Christopher Krauss. "Deep learning with long short-term memory networks for financial market predictions." *European Journal of Operational Research* 270.2 (2018): 654-669.

[39] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016.

[40] Chen, Tianqi, et al. "Xgboost: extreme gradient boosting." *R package version 0.4-2* (2015): 1-4.

[41] Zheng, Huiting, Jiabin Yuan, and Long Chen. "Short-term load forecasting using EMD-LSTM neural networks with a Xgboost algorithm for feature importance evaluation." *Energies* 10.8 (2017): 1168.

[42] Lim, Bryan, Stefan Zohren, and Stephen Roberts. "Enhancing Time Series Momentum Strategies Using Deep Neural Networks." *arXiv preprint arXiv:1904.04912* (2019).

[43] Dixon, Matthew F., and Nicholas G. Polson. "Deep Fundamental Factor Models." *arXiv preprint arXiv:1903.07677*(2019).

[44] Heaton, J. B., N. G. Polson, and Jan Hendrik Witte. "Deep learning for finance: deep portfolios." *Applied Stochastic Models in Business and Industry* 33.1 (2017): 3-12.

[45] Katsiampa, Paraskevi, Shaen Corbet, and Brian Lucey. "High frequency volatility co-movements in cryptocurrency markets." *Journal of International Financial Markets, Institutions and Money* (2019).