MATH 4995 Project 2 Report
G-Research Crypto Forecasting (Kaggle)

Leung, Ko Tsun (20516287),
Cheng, Tsz Yui (20595441),
and Yang, Po-Yen (20561878)
Kaggle Team : math4995_Leung_Cheng_Yang

1. Introduction

The cryptocurrency market has been skyrocketing recently, and it is estimated that over $40 billion worth of cryptocurrencies is traded every single day. Cryptocurrencies have become one of the most popular and trending assets for speculation and investment, however, it has been proven to be wildly volatile, where a person can make a fortune and become a millionaire in one day, and lose all his assets the day after. While a few people have made a great fortune through the fast-fluctuating prices, others have been experiencing losses. Hence, the objective of this project is to try whether we can predict some of these price movements in advance and forecast short-term log returns in 14 popular cryptocurrencies through machine learning techniques. Results will be evaluated in Pearson correlation coefficient between the predicted log return and the testing data in a 15-minutes difference.

2. Data Overview

2.1. Data Features

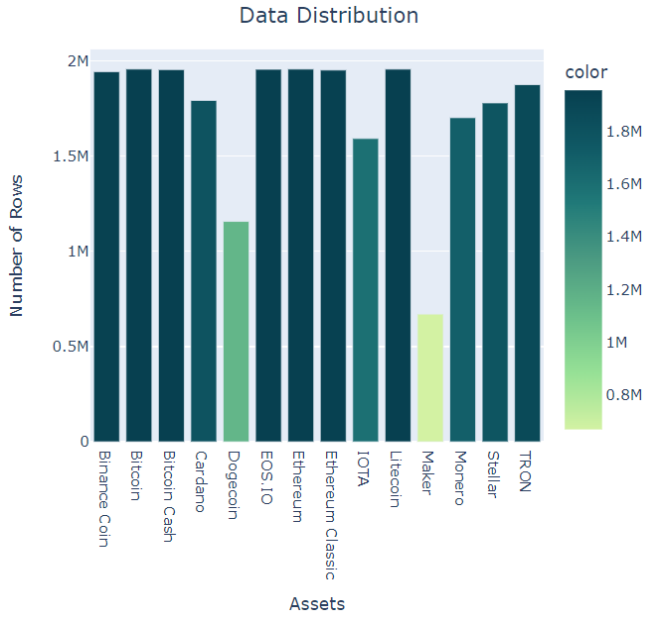
The data are time-series based, and each data consists of the following ten features:

Features	**Explanation**
Timestamp	All timestamps are returned as second Unix timestamps (the number of seconds elapsed since 1970-01-01 00:00:00.000 UTC). Timestamps in this dataset are multiples of 60, indicating minute-by-minute data.
Asset_ID	The asset ID is unique and corresponds to one of the cryptocurrencies (e.g. Asset_ID = 1 for Bitcoin). The mapping from Asset_ID to crypto asset is contained in “asset_details.csv”.
Count	Total number of trades in the given time interval (minute)
Open	Opening price of the time interval (in USD)
High	Highest price reached during the time interval (in USD)
Low	Lowest price reached during the time interval (in USD)
Close	Closing price of the time interval (in USD)
Volume	Quantity of asset bought and sold (in USD)
VWAP	Volume Weighted Average Price
Target	Residual log-return for the asset over a 15-minute timeframe

2.2. Exploratory Data Analysis

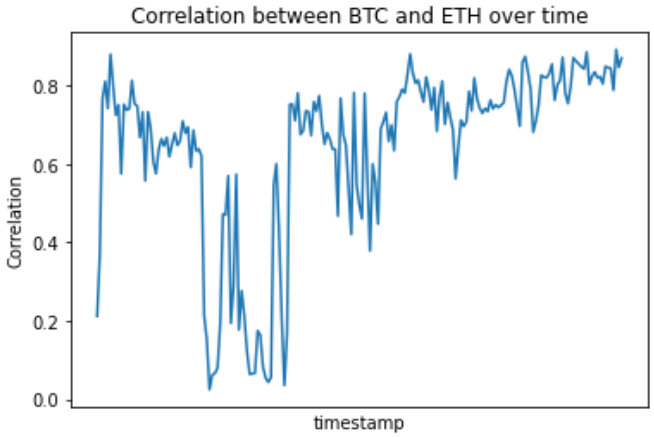
2.2.1. Data Distribution across different cryptocurrencies

We observe that most assets account for 1.5 million to 2 million in the training data. Dogecoin and Maker account for fewer data rows because they have not gained immense popularity until early 2021.

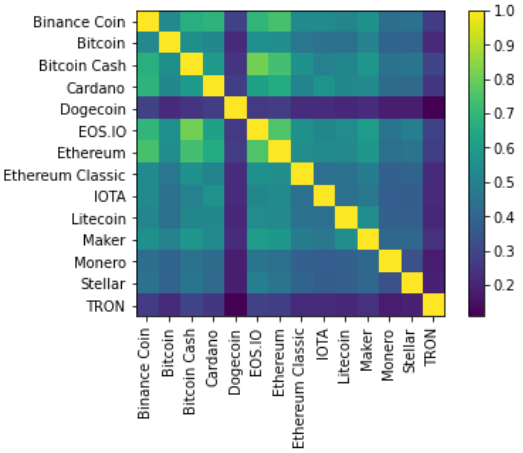


2.2.2. Correlation between cryptocurrencies

We hypothesized before that crypto-asset returns may exhibit some correlation. Take the correlation between Bitcoin and Ethereum during 2021 as an example:



We observe a high but variable correlation between the currency pair. There exists a non-stationary behavior in the correlation, characterized by a continuous change of statistical properties (mean, volatility, volume, etc.) over time. Stationarity is crucial since many useful analytical tools and statistical tests rely on it.



We also check the correlation across all the 14 cryptocurrencies by visualizing the correlation matrix. Note that there are some assets having a much higher pairwise correlation than the others.

2.3. Preprocessing

2.3.1. Fill In Missing Values

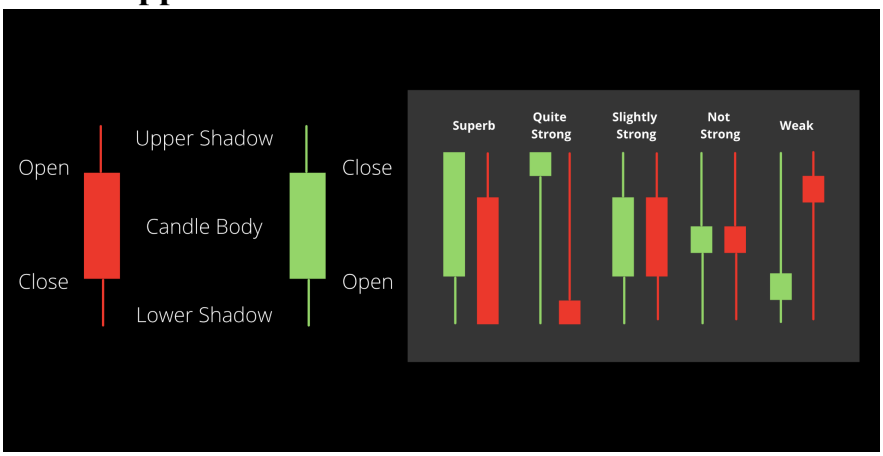
In this dataset, the target contains some NaN, and we filled these NaNs with 0.

2.3.2. Reindex

We have noticed that there are many gaps in the data. In order to work with most time series models, we should preprocess our data into a format without time gaps. Hence, we used the .reindex() method for forward filling, filling gaps with the previous valid value.

2.4. Feature Engineering

2.4.1. Upper/Lower Shadow



In technical analysis, the shadow (wick) of a candlestick is the thin parts representing the price action within the given timeframe as it differs from its high and low prices. The length and position of the shadow can help us gauge market sentiment in security.
We believe that a long shadow means the asset will reverse its current trend. For example, a tall upper shadow signifies strong buying action during the time interval, yet the fact that the closing price is much lower than the period high reveals that sellers successfully forced the price back down. Hence, it can be seen as a bearish signal.

2.4.2. Spread

Spread is the difference between period highs and lows. It is used as an indicator of volatility.

2.4.3. Mean trade

Mean trade is calculated as volume divided by the total number of trades in the time interval. It measures the market sentiment toward security. Generally, if the mean trade is extremely high, meaning a major institutional investor has traded a large volume of cryptocurrencies on the market. It is used as an indicator of this behavior.

2.4.4. Log Price Change

The log price change is the logarithm of the closing price divided by the open price of a specific asset in that minute. It can be used as the indicator of the price momentum of the asset.

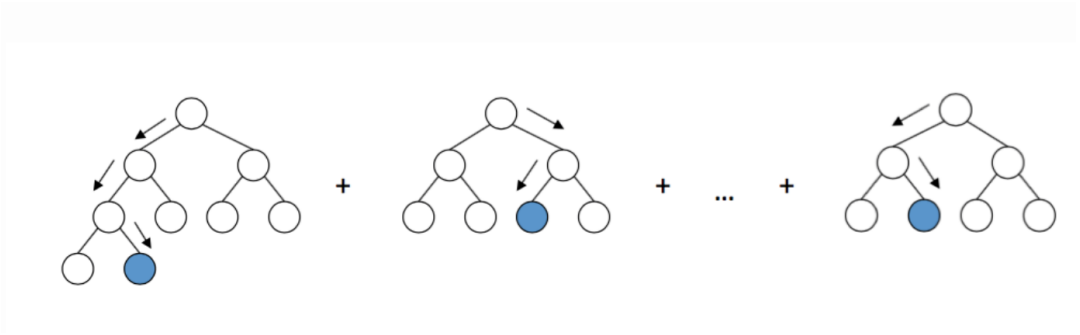
2.4.5. Absolute difference between closing price and VWAP

Volume weighted average price (VWAP) is a popular trading benchmark used by traders that gives the average price an asset has traded in a given time interval, based on both volume and price. For institutional investors, they aim at trading at VWAP as close as possible since it can reduce their cost to absorb multiple layers in the market order book. When the price deviates much from the VWAP, investors tend to buy or sell more to push the price back to VWAP.

3. Model Selection

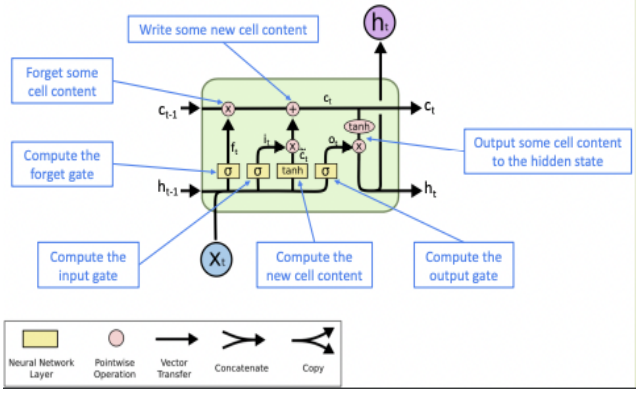
3.1. Light GBM

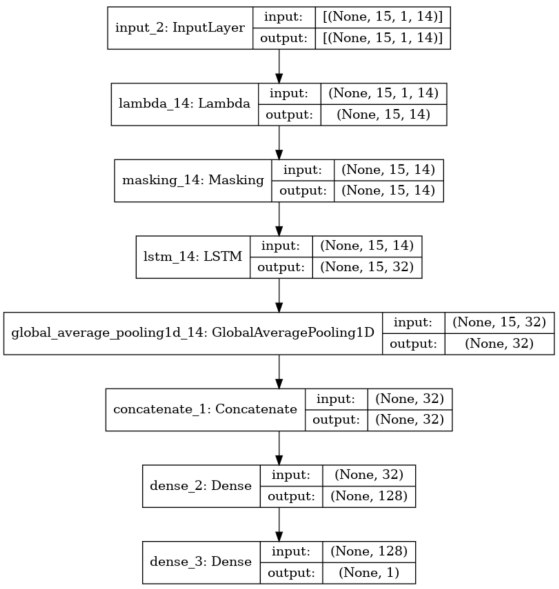
Light GBM is an efficient gradient boosting framework that uses tree-based learning algorithms. Light GBM grows its tree leaf-wise instead of depth-wise, meaning it chooses to grow the leaf with maximum delta loss. There are three reasons why we chose Light GBM to be one of our models. First of all, Light GBM has a high speed and also supports GPU learning. Second, Light GBM is able to handle data with a large size while taking lower memory to run. Last but not least, it focuses on the accuracy of results.
Leaf-wise growth of the tree is shown in the following image:



3.2. LSTM(Long Short-term Memory)

LSTM is a type of RNN proposed by Hochreiter and Schmidhuber in 1997. Compared to most RNN which uses a sigmoid function for the hidden layer function, LSTM adopts memory cells as shown in the following image:

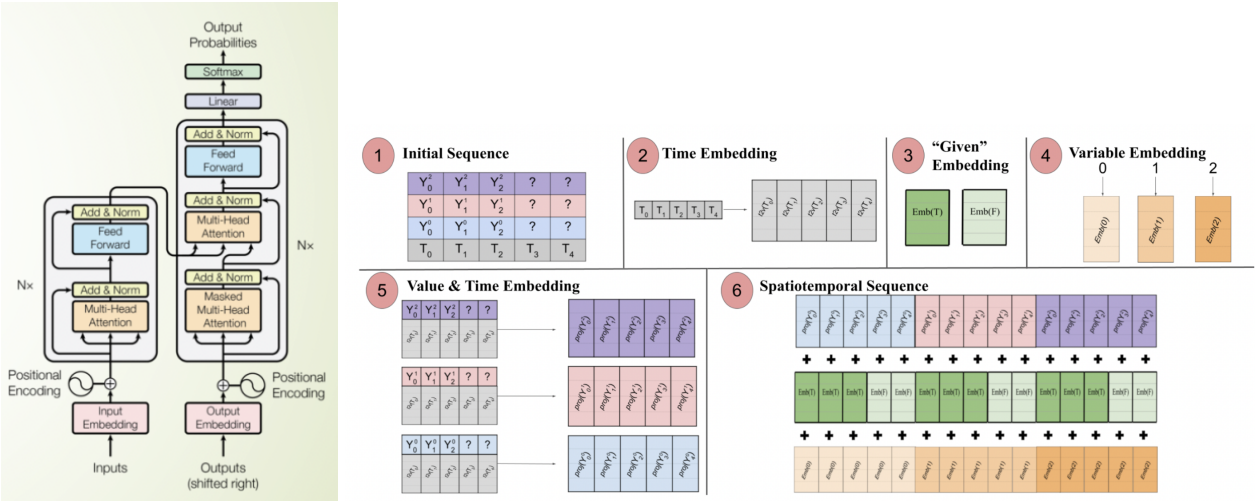




(The model detail is shown on the right). In addition, LSTM rectifies the short-memory problems from RNN. It manages to keep, forget or ignore data based on a probabilistic model. It can also be used to solve exploding and vanishing gradient problems. The reason that we chose LSTM as one of our models is that the cryptocurrency data that we are dealing with is time-series-based, and LSTM is well-suited in time-series-based data, as it can predict future values based on previous, sequential data.

3.3. Multivariate Time Series Transformer

Transformers are a state-of-the-art solution to Natural Language Processing (NLP) tasks. It is a deep model with a sequence of attention-based transformer blocks. We implemented the multivariate time series transformer model proposed in October 2020[1] and we extend the work to the cryptocurrency market. We choose this model as we want to examine the possibility of applying this state-of-art deep learning model to other time-series data domains.



Left: the structure of the transformer model. Right: We added the "spacetimeformer" in our model to output spatiotemporal Sequences from financial time-series input

3.4. Wavenet

Wavenet was originally proposed by DeepMind as a deep generative model of raw audio waveforms. It is popular among the NLP community as it can produce synthetic audio signals and it sounds more natural than other text-to-speech systems. It also showed good results in stock price prediction. Due to the similarity between stock price and the cryptocurrency market, we have made a hypothesis that Wavenet can also apply to the cryptocurrency market. We have applied some variation in the model to customize for the financial time series prediction task. Model details and hyperparameters are skipped here as it is too complicated, please refer to source code if interested.

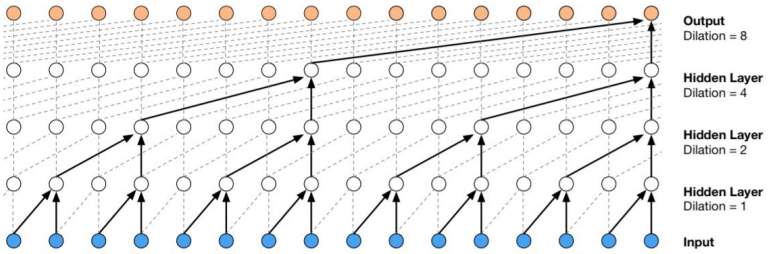


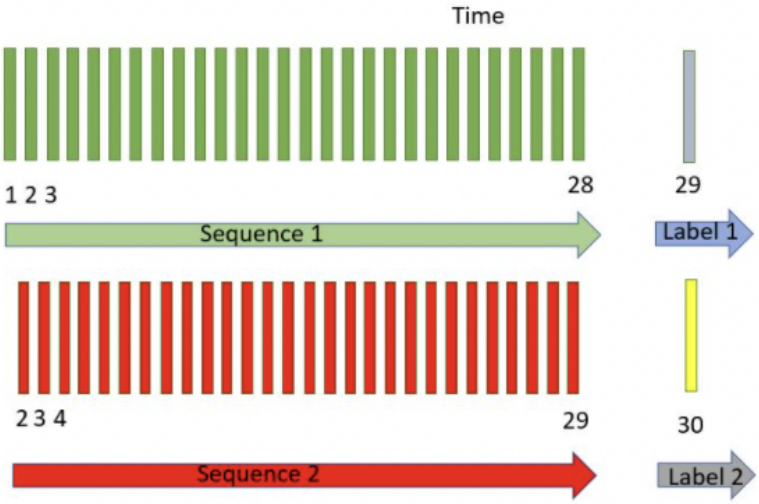
Figure 3: Visualization of a stack of *dilated* causal convolutional layers.

4. Model Techniques

Several modeling techniques are used to customize for this time-series-based problem. They can substantially improve our accuracy and reduce computation complexity.

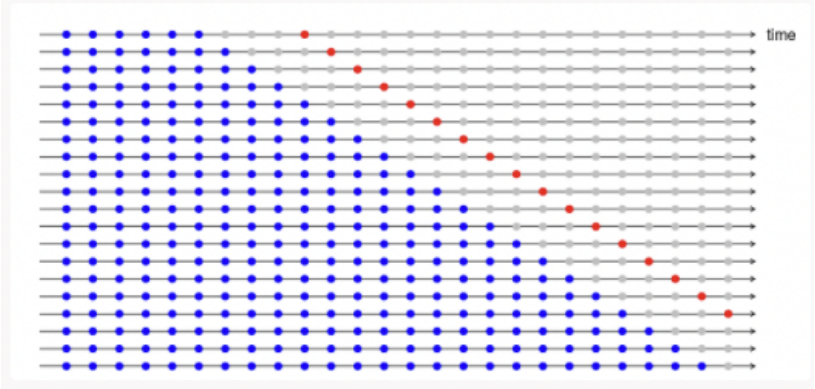
4.1. Time Series Sliding Window

The sliding window means using prior time steps to predict the next time step, and it is called a lag or lag method in statistics and time series analysis. While the number of previous time steps used is called the window width or size of the lag. In this project, we are dealing with multivariate time series, meaning two or more variables are observed at each time, and it is harder to model as often many of the classical methods do not perform well. Utilizing the sliding window, we can turn a time series dataset into sequences of 15 minutes and their labels, modeling the scenario to be a supervised learning problem, which is much easier to deal with.



4.2. Time Series Split Cross-Validation

The original version chooses random samples and assigns them to either the test set or the train set. However, when it comes to time series, we cannot apply the original cross-validation as it makes no sense to use the values from the future to predict values in the past. Hence, we have to apply time series split cross-validation, where the data are chosen in order and time series split divides the training set into two folds at each iteration on the condition that the validation set is always in front of the training set. In our analysis, we used the 10-folds cross-validation technique to train our models.

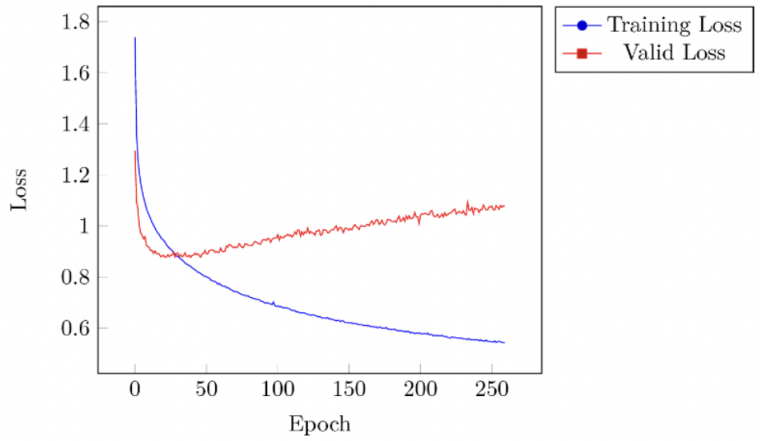


5. Result

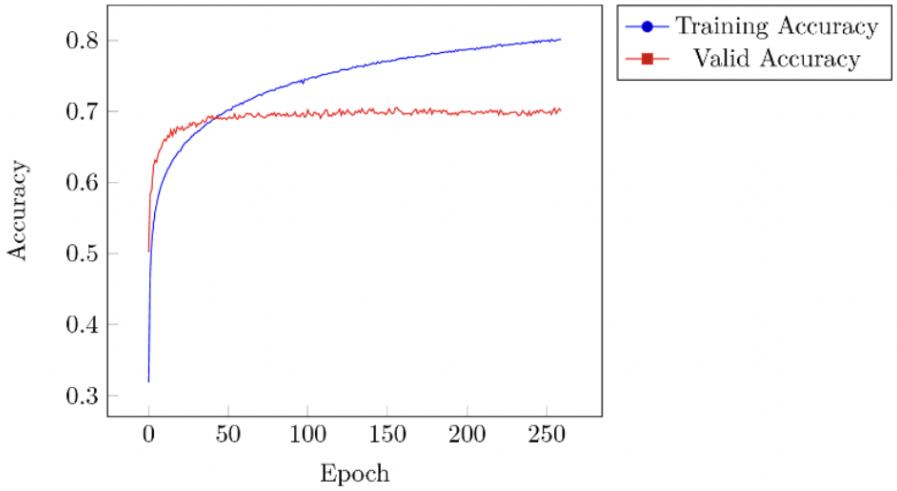
Model	Kaggle Score(Evaluation metric: Correlation)	Computation time
Light GBM	0.017	5032s
LSTM	0.467	10639s
Multivariate Time Series Transformer	0.539	14853s
Wavenet	0.695	36421s

6. Analysis

Wavenet is our most outperforming model(ranked among the top 30% teams) but it has the longest training time(~10 hours with K80 GPU). It proved that our hypothesis is correct, the deep learning model used in Natural Language Processing can also be applied to financial time series forecasting.



Mean squared error plot against epoch for the Wavenet model. The validation loss starts to increase after ~25 epochs. Therefore, early stopping is used in our later analysis to reduce computation effort and avoid overfitting.



The validation accuracy(Pearson correlation) converges to ~0.7 after 250 epochs. It can be observed that the Wavenet model can nicely capture the relationship in time-series data.

Moreover, it is important to notice that the accuracy of LightGBM is extremely low compared to other models. It shows that the Gradient Boosting Decision Tree methods are not suitable for the time-series contests, as it does not take in the consideration of input sequence.

7. Backtesting

We have performed a backtesting analysis to further validate our conclusion. We first start at $100,000 cash. A simple strategy is used: when a cryptocurrency is consecutively predicted to have a positive correlation in the next 15 minutes for 3 minutes, buy $5,000 dollars equivalent amount of this asset. Otherwise, sell the asset with the same amount. We tested this strategy on the portfolio of multiple assets for two years, with the weighting specified in the asset_details.csv provided by the competition organizer. It can gain 60% profit in 2 years from the start of 2018 to the end of 2019. It proved that our strategy successfully passed the backtesting analysis and it outperforms the price movement of the portfolio of cryptocurrencies.

Start	0.0
End	729.0
Duration	729.0
Exposure Time [%]	69.45
Equity Final [$]	353.5889
Equity Peak [$]	100000.0
Return [%]	60.646411
Buy & Hold Return [%]	-24.660214
Return (Ann.) [%]	0.0
Volatility (Ann.) [%]	NaN
Sharpe Ratio	NaN
Sortino Ratio	NaN
Calmar Ratio	0.0
Max. Drawdown [%]	-10.646232
Avg. Drawdown [%]	-10.646232
Max. Drawdown Duration	159.0
Avg. Drawdown Duration	159.0

8. Conclusion & Discussion

From the result, we can observe that Wavenet performed the best but it comes with the longest computation time, on the other hand, Light GBM performed the worst but with the lowest computational time, and this result is in accordance with our thoughts. Although Light GBM has a high speed and is able to process a huge amount of data, it is not suitable for predicting time series data, since all the data are somewhat correlated. As for Wavenet, it did perform well as we predicted since we thought that the stock price market is similar to the cryptocurrency market, however, due to the large computational cost, it is also the most time-consuming model.

We have the following thoughts on further improving the accuracy.
First, if we have more computing resources, we can try training different state-of-art deep learning time series models. Different variations can also be applied to the current models, including ensembling methods, data augmentation, etc.

Second, we can extend this idea to future cryptocurrency data, allowing it to automatically trade the assets. We may use stimulated positions first to avoid financial risks, and start to trade in real money if it is validated to have satisfactory performance on real data. We can also gain feedback from the actual performance of the model to design different trading strategies, so we can apply these experiences to improve our models.

Moreover, since this competition requires us to predict short-term returns for the next 15 minutes, we may extend this project to a shorter or longer term to observe different behaviors of the models if the prediction interval is changed. We can train a model by 5 minutes, 15 minutes, or 1 hour, or even extend it to 1 day. Hierarchical Time-series (HTS) can be used to achieve this goal.

9. References

[1] https://arxiv.org/pdf/2001.08317.pdf
[2] https://towardsdatascience.com/multivariate-time-series-forecasting-with-transformers-384dc6ce989b
[3] https://machinelearningmastery.com/time-series-forecasting-supervised-learning/
[4] https://medium.com/@soumyachess1496/cross-validation-in-time-series-566ae4981ce4
[5] https://www.kaggle.com/yamqwe/let-s-talk-validation-grouptimeseriessplit
[6] https://github.com/gzerveas/mvts_transformer

10. Contributions

Data Preprocessing & Exploratory Data Analysis	Cheng, Tsz Yui
Light GBM & LSTM	Yang, Po-Yen
Multivariate Time Series Transformer & Wavenet	Leung, Ko Tsun