

INSTITUT FÜR INFORMATIK
DER LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN



Master's Thesis

**On the topological property
of dynamic transaction graph**

Yuhao Wang

August 17, 2021

INSTITUT FÜR INFORMATIK
DER LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN



Master's Thesis

**On the topological property
of dynamic transaction graph**

Yuhao Wang

Aufgabensteller: Prof. Dr. Volker Tresp
Betreuer: Dr. Yunpu Ma
Abgabetermin: 17. August 2021

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 17. August 2021

.....Yukawa Wane.....
(Unterschrift des Kandidaten)

Abstract

With the explosion of the cryptocurrencies market, the market capitalization of cryptocurrency Bitcoin has already exceeded the global technology giant companies Facebook and Tesla in the beginning of 2021, taking the sixth place on the worldwide top market capitalization company list. At the same time, many global invest instituta announce to enter into cryptocurrency market, and massive different cryptocurrencies also has been playing more important role on the cryptocurrencies market, like Ethereum, its decentralized, open-source blockchain with smart contract functionality attract a lot of investition in the last few years. Undoubtful the decentralized and transparent features of the Bitcoin and Ethereum are their best trait to distinguish with the other market, but on the other hand their uncontrolled and unsupervised characteristics also led to the high risk of the investigation, and induce the money laundry. In order to optimize the market transaction of cryptocurrency, because of the entire transaction graph accessible to the public, based on the standard graph features this thesis will releavl the impact of the topological features with the price of the underlying cryptocurrency. In addition, the performance part of the simple feature prediction on the deep learning model is always lower than the traditional machine learning model. Through the development of the attention mechanism, the efficiency of the neueury network has gotten the glogist achievement in many areas. So we will also show the experiment of the application of the Informer model, a latest neueury network model development on Transformer.

Contents

1. Introduction	1
1.1. Background	1
1.1.1. Blockchain	1
1.1.2. Cryptocurrency Market	2
1.2. Outline	3
2. Data	5
2.1. Bitcoin Blockchain Dataset	5
2.2. Ethereum Blockchain Dataset	5
2.3. Cryptocurrency Market Transaction Prices	8
2.4. Data Preparation	9
2.4.1. Topological Features	9
2.4.2. Sliding Window	11
2.4.3. Feature Engineering	11
2.4.4. Feature Selection.	12
3. Methodology	17
3.1. Learning Graph Representations	17
3.1.1. ChainLet	17
3.1.2. ChainNet	18
3.1.3. ETH-ChainNet	19
3.1.4. Absorbing Random Walks (ARW)	20
3.2. Learning Topological Representations	21
3.2.1. Betti Number	21
3.2.2. Betti Sequence	22
3.2.3. Betti Derivatives	24
4. Machine Learning Models	25
4.1. Random Forest	25
4.1.1. Decision Tree	26
4.1.2. Random Forest Regressor	27
4.2. Gaussian Process	28
4.3. ElasticNet	29
4.4. XGBoost	31
4.4.1. Model Complexity	33
4.4.2. The Structure Score	33
4.4.3. Learn Tree Structure	33
4.5. Latest Neural Network Model: Informer	34
4.5.1. Encoder-Decoder Architecture	34
4.5.2. Canonical Attention Mechanism	35

4.5.3.	ProbSparse Self-attention Mechanism	39
4.5.4.	Self-attention Distilling	40
4.5.5.	Informer Model	40
5. Experiment		43
5.1.	Data pipeline	43
5.1.1.	Gather Raw Data	44
5.1.2.	Filtration and Aggregation	44
5.1.3.	Quantile Calculation	44
5.1.4.	Perseus Application	44
5.1.5.	Principal Component Analysis	45
5.2.	Parameter Setting	46
5.3.	RMSE of Models	47
5.4.	Performance of Topological Features	48
5.5.	Performance of Other Cryptocurrencies	49
5.6.	Comparison on Different Blockchain	50
6. Conclusion and Future Work		57
6.1.	Summary and Conclusion	57
6.2.	Future Work	57
A. appendix		59
A.1.	Feature Engineering	59
A.2.	Bitcoin time series data	59
A.3.	Performance Gain of all models	60
A.4.	Price prediction for 2016 with 1, 5, 10, 20 day horizons.	62
List of Figures		67
Bibliography		69

1. Introduction

Many previous studies have shown the importance of the basic features such as transaction amount and the previous price in predicting the bitcoin price. Moreover, as already observed in previous studies that reveal the relationship between the transaction amount, market price with the underlying graph structure, the topological data analysis (TDA) can capture these higher-order interactions[AAG⁺19]. However, in the last year, due to the rapid development of the cryptocurrency market, the market price of Bitcoin transactions has increased over sixfold, and massive large-scale international capital has poured into this market. In addition, the blockchain technology-based on Bitcoin has been gradually optimized, and a variety of new cryptocurrencies with different characteristics are constantly being introduced to the market, such as Ethereum, Litecoin, and Dogecoin. Therefore, the general strength of TDA in the analysis of various cryptocurrency transactions urgently needs to be further verified.

In contrast, different researches have shown the high performance of the vanilla Machine Learning algorithms such as Random forest, Gaussian Process, and ElasticNet in the long term bitcoin price prediction. However, the latest efficient deep learning model in time series data prediction like RNN, LSTM, and other neural networks did not significantly improve[AAG⁺19]. Since 2017 with the introduction of the attention mechanism in neural networks, the vanilla transformer's performance has achieved satisfactory results in many research fields [VSP⁺17]. Then last year, based on the vanilla Transformer, Haoyi Zhou proposed a more efficient model, Informer, which successfully enhanced the prediction capacity in the LSTM problem and used the ProbSparse Self-attention mechanism to replace the canonical self-attention for the reduction of the time complexity and memory usage[ZZP⁺20].

In this research, we focus on topological properties on the dynamics graph of cryptocurrencies of Bitcoin and Ethereum and use the latest attention neural network model Informer to improve the price prediction of cryptocurrencies. At last, we will also explore the different influence of the topological properties on the various cryptocurrency datasets based on their individual characteristics.

1.1. Background

1.1.1. Blockchain

A typical block is composed of multiple transactions and other related information. All these blocks are combined one by one in a time series according to the hash value to become a blockchain. Hence it also represents a complete ledger of the transaction history [Nak19]. Blockchain can be validated by the dynamic graph using cryptographic means. Each block includes a hash number of the last block, a random number which is used to verify the hash value, and a timestamp [Nak19], figure 1.1 is the basic structure of blockchain. Besides, each substantial slight change of a block would reflect the respective hash value computation. This mechanism can effectively prevent all fraud behavior in transactions on the blockchain. In addition to the rules and procedures of the extension of the blockchain, each transaction recorded in the ledger must be stored in a block. For example, Bitcoin

1. Introduction

generates a block with all transactions every 10 minutes by miners who validate the blocks and obtain the reward with Bitcoin. With this cryptograph, all customers can establish a reliable network to transfer assets without a central institution. In contrast to traditional centralized systems, the distributed system has many disadvantages for system persistence and user data security. The system does not depend on a specific intermediary or other participants, and there are no third parties to collect personal data.

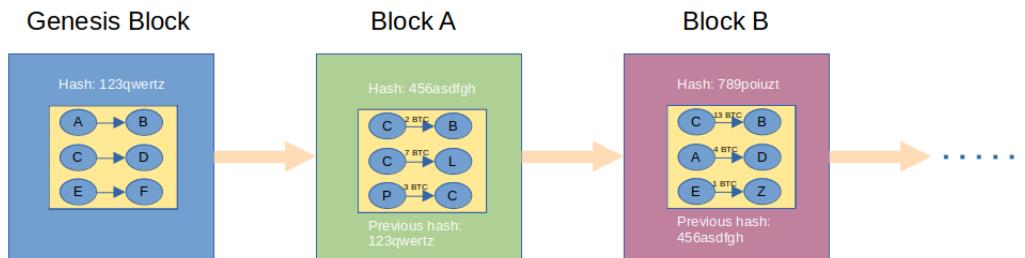


Figure 1.1.: An example of a blockchain composed of the genesis block and blocks A, B and other blocks. The system generates a new block every 10 minutes.

In recent years, various papers and new technology around the blockchain have been published. One of the essential derivative concepts undoubtedly is the combination of the blockchain with "Smart Contracts," which are also the cornerstone of Ethereum, another popular decentralized cryptocurrency first proposed by Buterin in 2014 [B⁺13] [ZZWD20]. The original concept of "smart contract" suggests establishing a contract through a combination of computer protocol and user interface. Depending on the latest development of blockchain technology, its more accessible utilization feature makes it possible to replace lawyers and banks with predefined aspects. As an extension of the Bitcoin blockchain, Ethereum can be applied in a broader scope. The contract established by cryptography can replace the notary and other third parties and offer many benefits, such as a cost-effective, transparent, and secure manner. With the coming evolution of the blockchain, many financial industries try to implement blockchain technology in their current business to reduce their transaction redundant and save on the cost of the procedure [ZZWD20].

1.1.2. Cryptocurrency Market

Due to the surging institutional demand, from February to April in 2021, the total market value of the cryptocurrencies has doubled, and for the first time, its number past \$2 trillion. In contrast to Bitcoin, the largest of the cryptocurrencies, its price six-fold in the single last year. At the beginning of February 2021, Tesla boss Elon Musk announced that he would accept Bitcoin as a means of payment in the future, and the manufacturer had bought Bitcoin worth 1.5 billion US dollars. With the acceptance of the cryptocurrency, Tesla's Bitcoin portfolio is also growing. In addition, MicroStrategy, Square, and other global companies are the largest aggregate investors in the last several years. Since the explosive growth of investors' attention and interest in cryptocurrencies, Bitcoin's market value has surpassed Facebook and Tesla this year, ranking sixth in the global corporate market capitalization ranking. In addition to Bitcoin, Ethereum has also received a large amount of investment in the past year, which has prompted the market capitalization of Ethereum to increase by 20 times in the short term. In contrast, the overall cryptocurrency market has exceeded 2 trillion in the past year, and the current types of cryptocurrencies have exceeded 660, according to

the latest market statistics and research. With the exploration of cryptocurrency and the improvement of blockchain technology, cryptocurrency is gradually developing into a stable and reliable market investment product. It obtains more support and truthfulness from global investors. Figure 1.2 shows the ranking of top cryptocurrencies Bitcoin and Ethereum in the world capital market [com].

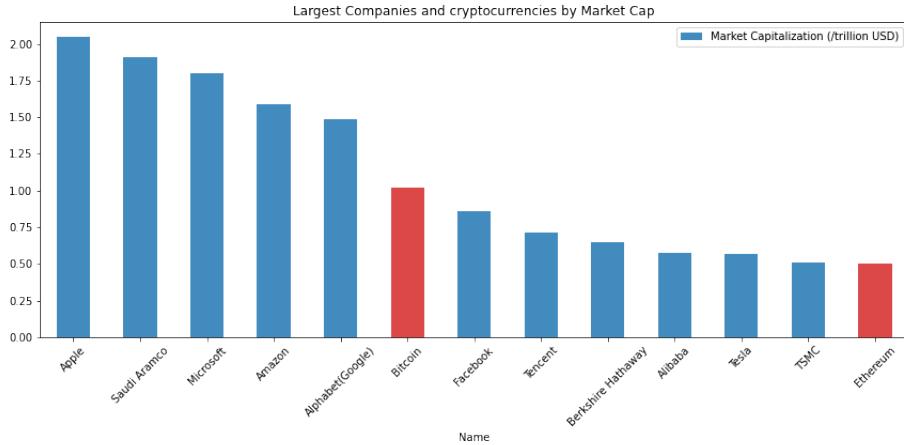


Figure 1.2.: The ranking of top cryptocurrencies Bitcoin and Ethereum in the world capital market.

1.2. Outline

Chapter 1 gives the background information of the blockchain, and the market capitalization development of the cryptocurrencies such as Bitcoin and Ethereum are introduced. Also, the frequently used terms and definitions are listed in this chapter.

Then in chapter 2, the blockchain dataset of different cryptocurrencies are expressed, based on the who-trust-who traction network, the original blockchain dataset of bitcoin can directly download from bitcoin application, or it can be extracted from the official blockchain dataset analysis API. In this section, there is also an explanation on how to extract the valuable features from the massive data. Furthermore, because the complete Ethereum blockchain dataset has already exceeded 1 TB, a part of dataset capture is through the Infura service. Moreover, the total market price of Bitcoin and Ethereum is downloaded from an online source. However, all the dataset is given with the original type and includes prominent useless features, so the necessary feature engineer extracting some statistical properties will be explained in this part.

The third chapter is the essential part of this article, and the first section will primarily express the graph representations of the cryptocurrencies. Then the concept of the topology is introduced in the second section, such as Betti number, Betti sequence, and other topological features. After the process of the data, some traditional machine learning models and the explanation of their parameter selection are given. The last phase of this chapter was the details of the step-by-step from attention mechanism to the latest neural network with ProbSparse self-attention mechanism Informer. In addition, this section expresses the procedure of Informer processing different blockchain data and predict the price of Bitcoin and Ethereum in specific periods.

Then the parameter setting, experimental results, benchmark, and comparison are in chapter 4. The detail of the experimental data set, model parameter sets are explained in the first phase. The

1. Introduction

experimental results of various cryptocurrencies on different machine learning models are introduced in the second part. The evaluation of the results primarily includes RMSE and Performance Gain of machine learning models. The last section of this chapter compares and analyzes the results for different types of blockchain datasets of Bitcoin and Ethereum on the topological features and machine learning models.

The last chapter includes conclusion and feature work, expressing some rational analysis conclusions based on the processed results and some suggestions in this direction for following step study opportunities.

More experimental details obtained from capturing and processing the original cryptocurrency blockchain data set are given in the appendix. It shows in detail the impact of cryptocurrency and topological features and different machine learning models.

2. Data

Because of the Bitcoin Blockchain's open-source feature, all of the blockchain's original data can be downloaded by anyone with access to the Internet. Through the official Bitcoin software, users can download the whole dataset of the Bitcoin Blockchain. But in the current thesis, a different method was chosen in which the dataset was gathered through the API, provided by Blockchain.com and Web3. As the original whole blockchain dataset is redundant, after gathering the dataset from specific periods through the API, the python script was employed to filter out the useful experimental features, efficiently saving the hardware and time cost of the experiment.

2.1. Bitcoin Blockchain Dataset

In the present research, the entire Bitcoin transaction graph from January 2009 to April 2021 was downloaded via the API of blockchain.com, which also possesses the aggregated price of all online exchanges. The most used ordinal data in previous research conducted on Blockchain and cryptocurrency are rudimentary and focus on superficial features of Bitcoin, such as the total transaction amount, transaction volume, or the cryptocurrency's previous market price. Regarding the details of the data gathered in the present experiment, a python script was designed to request the single block information. Firstly, the date was converted to a Unix number and then the daily blocks with the Unix number were requested. Subsequently, a loop function was utilized to request each block's information from the last step and filter the specific features from the large datasets. Finally, the daily input and output degrees were aggregated on the transactions of each block.

An example of the raw block data obtained from blockchain.com's API is exhibited in Figure 2.1. The Bitcoin system generates a block every 10 minutes; predicated on the Blockchain's service, anyone can obtain any block of information with a specific Unix DateTime. Each block consists of several transaction data similar to that articulated in Figure 2.2, and from this, the input and output size and other transaction features can be extracted for further experimentation.

2.2. Ethereum Blockchain Dataset

Ethereum is the most substantial open-source and decentralized blockchain-based cryptocurrency on the smart contract functionality, with its market capitalization being ranked second of all cryptocurrencies. Vitalik Buterin proposed the concept of Ethereum in 2013, and the genesis block originated on the 30th of July 2015 with 72 million ETH coins as the initial supply[B⁺13]. Presently, a series of upgrades have been implemented into Ethereum 2.0, such as utilizing sharding to increase transaction throughput and a transaction to proof of stake has also started being implemented. Until now, the volume of the complete Ethereum blockchain dataset exceeds 1TB. To optimize procedure time and storage space, the python web3 package and INFURA [Inf] services extract Ethereum Blockchain original datasets. The specific steps of the data extraction are similar to the Bitcoin Blockchain procedure.

2. Data

```
hash: "0000000000000000222201a85fe8e41b126cfb659927138a1b8ba4fec2c166f"
ver: 536870912
prev_block: "00000000000000001cb325544b07decf00227beef7b3aa0d551c8b8da17d13d"
mrkl_root: "c34a520fc12d9fb779a538074b1ae6b53a6e5932cb74a5ba30e5bdeb38137b6d"
time: 1483223893
bits: 402879999
► next_block: [] 1 item
  fee: 17475328
  nonce: 3724022291
  n_tx: 862
  size: 398962
  block_index: 446025
  main_chain: true
  height: 446025
  weight: 1595848
  ► tx: [] 862 items
```

Figure 2.1.: An example for BTC raw block data from blockchain.com.

```
▼ tx: [] 862 items
  ▼ 0:
    hash: "3978246c29ecbdc06695072bbd4136be66bc8815feb0bd563c7646f863cc985c"
    ver: 1
    vin_sz: 1
    vout_sz: 1
    size: 127
    weight: 508
    fee: 0
    relayed_by: "0.0.0.0"
    lock_time: 0
    tx_index: 3257962684418254
    double_spend: false
    time: 1483223893
    block_index: 446025
    block_height: 446025
    ▼ inputs: [] 1 item
      ▼ 0:
        sequence: 4294967295
        witness: ""
        script: "0349ce061d4d696e656420627920416e74506f6f6c2077792031770f87205868335553020000c1610300"
        index: 0
        prev_out: null
    ▼ out: [] 1 item
      ▼ 0:
        type: 0
        spent: true
        value: 1267475328
        ▼ spending_outpoints: [] 1 item
          ▼ 0:
            tx_index: 5369957813877720
            n: 26
            n: 0
            tx_index: 3257962684418254
            script: "76a9149160230dc0b0662088d2842aafeb7c09912cfb6c88ac"
            addr: "1EFg9XXX1U99pNJJTeQwuuEpbHFW4XS8uL"
  ▶ 1:
  ▶ 2:
```

Figure 2.2.: An example for BTC transaction data from blockchain.com.

Figure 2.3.: An example for ETH raw block data from Web3 API.

```
AttributeDict({'blockHash': HexBytes('0x191fdbd0dadf13b2e406b23da256d6004405ceb19fe60b2a  
'blockNumber': 11920000,  
'from': '0x3f5CE5FBFe3E9af3971dD833D26bA9b5C936f0bE',  
'gas': 207128,  
'gasPrice': 177000000000,  
'hash': HexBytes('0x43a1f5650e72c72b4c7b8e8f7fd81445688f187f9e11ef91c7011a2c6b5ae2c0'),  
'input': '0xa9059cbb00000000000000000000000000000004faa915826902c21bc0af2201297608aa6a7c74d000  
'nonce': 6750836,  
'r': HexBytes('0x362cb63f46708cdc90ce415f239f2ddbd35808c592a4ec4c0c46bed7f575aa24'),  
's': HexBytes('0x09d6db0b0e23367347640eeb807ab0cb7b4775562c0523f0f6767195db708d8f'),  
'to': '0xdAC17F958D2ee523a2206206994597C13D831ec7',  
'transactionIndex': 0,  
'type': '0x0',  
'v': 38,  
'value': 0}),
```

Figure 2.4.: An example for ETH transaction data from Web3 API.

Although the formulation of Ethereum is predicated on Blockchain technology, it is comprised of different structures and brilliant contracts. Additionally, as exhibited in Figure 2.4, Ethereum possesses no input and output sizes as Ethereum transactions involve a single source and a single target account. On the other hand, Figure 2.3 articulates how Ethereum also employs a timestamp to mark each block. However, the Ethereum development team does not provide any service for

2. Data

users to obtain block data via timestamp or Unix date. As a result, the block number must be used to request block data one by one and then merge them into daily block data in accordance with their timestamp. The total quantity of Ethereum blocks in 2021 has exceeded 12 million, with its volume still rapidly growing. Despite it being a daunting challenge, current research on cryptocurrency aims to formulate a stable and efficient program to process this explosive data.

2.3. Cryptocurrency Market Transaction Prices

The time-series Bitcoin and Ethereum prices are extracted from investing.com, along with the complete Ethereum dataset and a part of the Bitcoin dataset. Each record of the website provides a timestamp in Epoch and the exchange rate in the dollar.

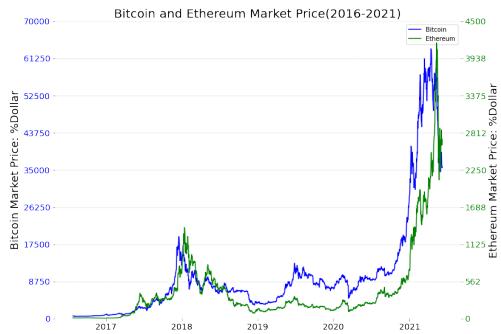


Figure 2.5.: Bitcoin and Ethereum Market Price(2016-2021).

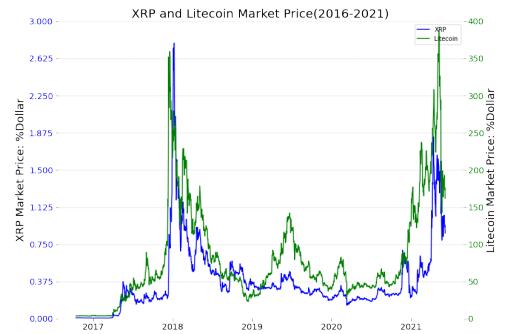


Figure 2.6.: XRP and Litecoin Market Price(2016-2021).

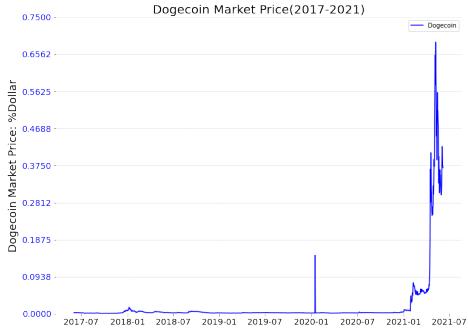


Figure 2.7.: Dogecoin Market Price(2017-2021).

Predicated on the price collected from the investing.com platform, Figure 2.5 displays how the Bitcoin and Ethereum prices have widely fluctuated in late 2017. During this period, the Bitcoin price peaked at nearly 20,000 dollars, and the price of Ethereum was also almost 1380 dollars. People labeled this period the ‘Cryptocurrency Bubble’[Sch19]. Preceding, the first week of 2018 witnessed

the massive spike in the price of Bitcoin and Ethereum. However, with the sudden emergence of COVID-19 at the beginning of 2020, the price of cryptocurrencies has continuously risen over the course of one year. As of today, nobody can predict the end of this rise and what will happen if there is another cryptocurrency market price crash.

In addition to the most popular cryptocurrencies - Bitcoin and Ethereum - the capital market accommodates a significant variety of coins. The present research also employs Bitcoin-derived XRP, Litecoin, and the currently trending Dogecoin to analyze the economic changes to the price of cryptocurrencies and their market capitalization. Through the observation of the price changes in different cryptocurrency markets, the mutual influences between different cryptocurrencies are shown to serve a critical role in exploring the future development trend of cryptocurrencies. Figure 2.6 articulates the clear correlation between the market price of XRP and Litecoin. Furthermore, Figure 2.7 demonstrates the dramatic fluctuations in the market price of Dogecoin in the last year. With the emergence of new types of cryptocurrencies, various currency trading platforms, such as Binance coin, Cardano, and other institutions, are also constantly arising.

Recently, Amir elaborated upon various structural properties of the transaction graph of several cryptocurrencies on the market and detailed the relationship between the quantity of monthly edges and nodes for the non-cumulative transaction graph with their market price [MB19]. The aforementioned research utilized the transaction graphs of Bitcoin, Litecoin, Ethereum, Dash, and Z-Cash.

On the other hand, with the development of the cryptocurrencies trades, various cryptocurrency exchange platforms are arisen, like Binance, and Cardano. Binance is a Cayman Islands-domiciled cryptocurrency exchange that provides platforms for trading various cryptocurrencies founded in 2017. As of April 2021, Binance was the largest cryptocurrency exchange in the world in terms of the trading volume [Coib]. Throughout Binance's history, this organization has provided two self-developed cryptocurrencies: Binance Coin (BNB) and Binance Smart Chain[Tap]. The characteristic of Binance Smart Chain is "Proof of Equity", which combines proof of authority and proof of equity. Currently, Binance has 21 approved validators, and as of 2021, Binance Coin is the third-highest cryptocurrency by market capitalization[Tap]. Besides, the user of Binance can pay fees in transactions with BNB[Tap].

2.4. Data Preparation

Predicated on the entire Blockchain data of Bitcoin and Ethereum starting from day one to 2020 December, the blockchain graph was built with an interval of 24 hours from 2017 and 2020. In addition to Bitcoin and Ethereum blockchain data, we also scraped the market prices of various cryptocurrencies against the U.S. dollar from the Internet. These cryptocurrencies including Bitcoin(BTC), Ethereum(ETH), Litecoin(LTC), Dogecoin(DOGE), XRP, Binance(BNB), Tether(USDT), and Cardano (ADA). The original data source of Bitcoin transactions is Blockchain.com, for raw Ethereum transaction data was provided by Infura.io, then the historical market price of all cryptocurrencies to the dollar was scraped from investing.com.

2.4.1. Topological Features

Filtration data: The Blockchain data is employed to form the occurrence and amount matrix with the dimension $N \in \{5, 10, 20, 40\}$. Because N of 5(*i.e.*, $\mathbb{C}_{i \rightarrow os.t., i \leq 5 \text{ and } o \leq 5}$) covers 90.50% *chainlets*, and N of 27 covers 97.57% *chainlets* [AAG⁺19]. As a result, N = 20 was selected to establish the occurrence matrix, which involves 400 chainlets and provides a dense matrix.

2. Data

Betti number and Betti Derivatives Data: For this calculation, Perseus software was used [Nan] - a popular tool for calculating the Betti numbers [MN13].

After the consideration of the computational costs of the data processing, the one-dimension Vietoris-Rips (VR) complex was calculated during the present experiment. This calculation means that the loop is formed by more than 2 nodes and leads to Betti-0 and Betti-1 negatively. When ϵ is bigger, the complex possesses more simplices. Simultaneously, the complex possesses less connected components and more loops. The comparison Figures 2.10 and 2.11 detail the rise of the average Betti-0 curve matches and the fall of the corresponding Betti-1 curve and vice versa. On March 27th, 2020, the Coronavirus Aid, Relief, and Economic Security Act (CARES Act) [Gov] was passed by the U.S. Government, and \$2.2 trillion economic stimulus entered capital market. As a result, an immediate rise wave of the whole cryptocurrency market was set in motion. With the subsequent follow-up of the economic stimuli by governments around the world, the peak of Bitcoin's soaring price has been maintained until the end of 2020.

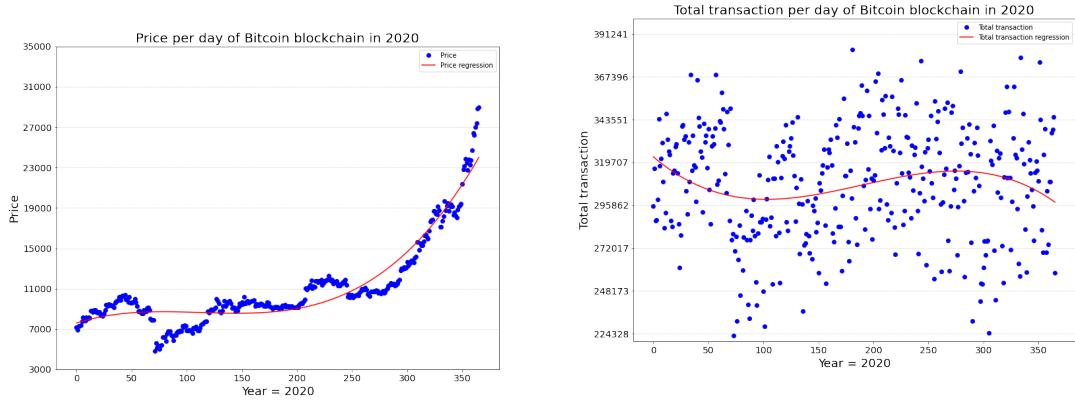


Figure 2.8.: Price per day of Bitcoin in 2020.

Figure 2.9.: Total transaction per day of Bitcoin blockchain in 2020.

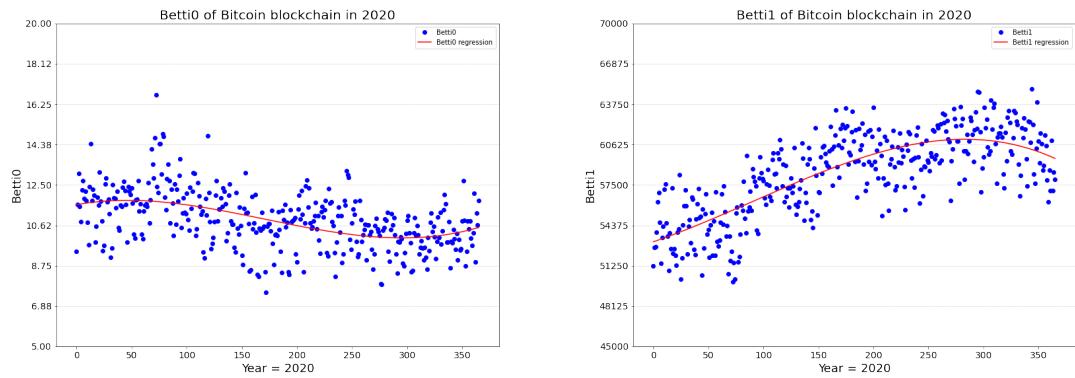


Figure 2.10.: Betti0 of Bitcoin blockchain in 2020. Figure 2.11.: Betti1 of Bitcoin blockchain in 2020.

2.4.2. Sliding Window

The overall cryptocurrency price in the last several years can be perceived as a time series data. In the present investigation, the sliding window was employed to predict the point of the price. To illustrate, when the horizon $h = 1$ and window $w = 1$, the model employs the data of the last 1 (window) day to predict the price of the next 1 (horizon) day, then the window size w denotes training the model with the last w days to predict the price of the h -th day in the future. Algorithm 1 exhibits the procedure of the sliding window:

Algorithm 1 Sliding window

Input: Data: $\{(x_t, y_t) : t \in T\}$ where $x_t \in \mathbb{R}^d$; y_t : the daily bitcoin price in dollars; l : training length; w : sliding window length; h : prediction horizon; d_2 : pca dimension.

Output: θ : Model Parameters.

```

 $x_{train}, \hat{x}_{train}, y_{train} \leftarrow \{\}$ 
for each  $t \in [h + w : l]$  do
     $\hat{x}_t \leftarrow [x_{t-w-h+1}, \dots, x_{t-h}; y_{t-w-h+1}, \dots, y_{t-h}]$  ▷ row-wise
     $\hat{x}_{train} \leftarrow \hat{x}_{train} \cup \hat{x}_t$ 
     $\hat{y}_{train} \leftarrow \hat{y}_{train} \cup \hat{y}_t$ 
end
 $x_{train} \leftarrow PCA(d_2, \hat{x}_{train})$ 
 $\theta = model.fit(x_{train}, y_{train})$ 
return  $\theta$ 

```

The window size $\{3, 5, 7\}$ and horizon size $\{1, 2, 5, 7, 10, 15, 20, 25, 30\}$ of this research were selected in accordance with the experiment parameter suggestion of the paper ‘ChainNet’. Subsequently, the models were trained with the dataset in length $l \in \{100, 200, 300, 500\}$. After considering the nonlinear effects of the interaction between the horizon, window, and training length, the grid search of all the parameters was utilized to filter out the parameters that result in the best prediction.

Furthermore, as the most crucial part of the model training with a sliding window approach, each machine learning model was trained constant times per each prediction, this setting has efficiently enhanced the results of the batch prediction approach. Regarding the vanilla machine learning model with low dimensional features, this setting undoubtedly does not cost much hardware and time resources. But, when discussing the time cost of the latest neural network model, there is still substantial room for improvement.

Figure 2.12 is the demonstration of the sliding window, the blue line refers to the market price of Bitcoin in the US dollar in 2020.

2.4.3. Feature Engineering

The most significant achievement of feature engineering is the substantial enhancement of prediction making, as feature engineering allows the useful pattern of the original data to be extracted. Additionally, the capability of feature engineering is not only a subjective process, but also highly dependent on deep domain knowledge.

The most involved features in network research are window features and node-based features. Window features primarily include the average transaction value and the transaction amount in the network, referring to the data on the whole network level along with the data on the level of both networks, neighborhoods or nodes, such as the in and out degree. Due to the Node-based features’

2. Data



Figure 2.12.: Visualization of a sliding window, with window size w and h the prediction horizons.

vast consumption of computing resources, this feature is rarely employed in machine learning projects. Despite this, the transactions between the users decide the cryptocurrency market price, and the node-based features explore the relationship on a more in-depth level between users.

2.4.4. Feature Selection.

During preparation, the most important features of the mutual information between the most basic features and cryptocurrencies were filtered out via appropriate calculations; the greater the mutual information between the most basic features and cryptocurrencies, the stronger the correlation between them.

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log\left(\frac{p_{(X,Y)}(X, Y)}{p_X(x)p_Y(y)}\right)$$

In the formula, $p(X, Y)$ refers to the joint probability and p_X, p_Y denotes the marginal probability mass for the two elements X and Y. In the present experiment, the sklearn package of *metrics.mutual_info_score* in Python was the tool utilized when calculating their mutual information. Based on the primary properties extracted from Bitcoin and Ethereum raw transaction data.

Feature selection not only functions to reduce training time, but fewer kernel features also enhances accuracy as a large quantity of useless features burdens the overall enhancement of machine learning performance. In addition to the prior superficial influence, excessive features leads to more variance and causes overfitting of the trained machine learning model. Though many features are related to the price of Bitcoin to some extent, such as price, mean degree of addresses, quantity of new addresses, mean and total coin amount transferred in transactions and address network average clustering coefficient, price and total transaction amount(TotalTx) were observed to be the most useful features and are the only basic features employed for training machine learning models[AAG⁺19, RS13].

However, in table 2.1 the mean volume shows the greatest impact against Bitcoin price, it's mutual information with Bitcoin price reaches 0.52, the mutual information of the total transaction

Table 2.1.: Mutual information of all blockchain features and various cryptocurrencies.

BTC features	Mutual information (I)	Cryptocurrencies	Mutual information (I)
Mean volume	0.52	ETH price	1.9
Median volume	0.25	LTC price	1.6
Std Volume	0.44	XRP price	1.6
Sum Volume	0.49	DOGE price	1.3
Total transaction amount	0.21	USDT	0.39
Input degree	0.1	BNB	1.5
Output degree	0.14	ADA	1.7

amount with Bitcoin price has only 0.21. Though this represents a stronger correlation that exists between the property of the mean volume and cryptocurrency price, the later experiment has not found a significant difference between the result of mean volume and total transaction amount. For the consistency of the experiment and to simplify the experiment steps, we continue to choose the total transaction amount(TotalTx) as one of the parameters of the experiment.

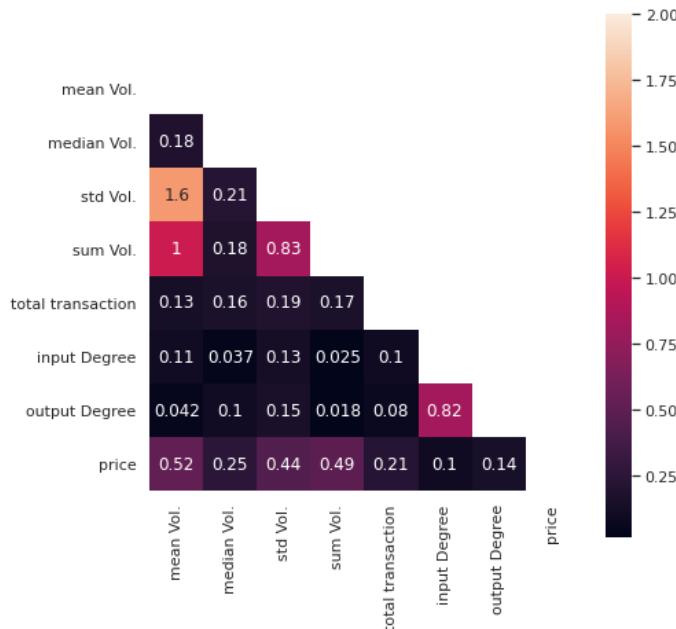


Figure 2.13.: The covariance matrix of Bitcoin blockchain various properties.

Due to the restrictions of the mutual information, only one-to-one relationships between a feature and the cryptocurrency price can be investigated. When two features correlate with the different parts of the cryptocurrency price and possess high covariance, but their mutual information is not significant when compared with the cryptocurrency price, the combination of them may provide a higher contribution when optimizing a prediction.

2. Data

Regarding the details of Bitcoin, Figure 2.13 exhibits high covariance between the mean volume, standard volume, and sum volume of the transaction. Most dark grids in Figure 2.13 express the faint correlation between most blockchain properties except the mutual information between input and output degree. This result also reveals the potential investigative value of input and output degrees on the blockchain transaction graph. However, the fields of input and output degree are almost obsolete since their value is similar for all BTC interactive features, making the superficial input and output degree insignificant for the price of BTC.

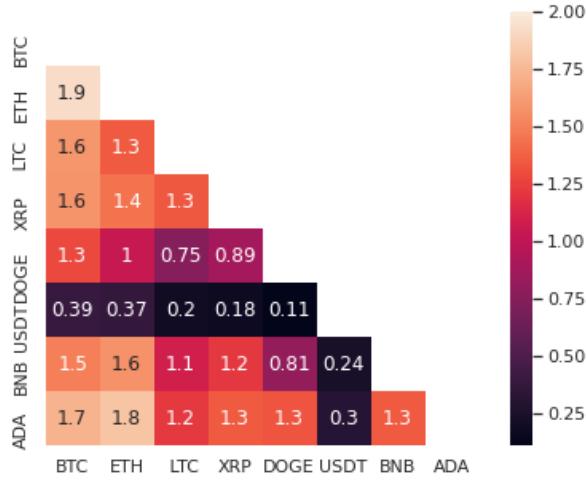


Figure 2.14.: The covariance matrix of the top 8 cryptocurrencies price by market capitalization.

In Figure 2.14, due to the rapid development of ETH in recent years, the price of Ethereum (ETH) is illustrated to serve as the most significant influence on the price of Bitcoin (BTC) - the covariance between the price of ETH and BTC equating to 1.9. With additional new technologies and innovation in ETH, ETH has become Bitcoin's strongest competitor. On the other hand, the market price of Litecoin, XRP, Dogecoin, Binance, and Cardano also yield excellent results of mutual information with Bitcoin price, only USDT displays a significant difference with the most cryptocurrencies. This discrepancy may be caused by the investigation of the New York Attorney General, who announced the end of the illegal activities of the virtual currency trading platform Bitfinex in New York [GEN21], which absolutely crushed the trading market of USDT, and the confidence of the cryptocurrency investor.

Different from the previous blockchain features, the large area of orange in Figure 2 shows that most cryptocurrencies have a strong correlation with the market price of Bitcoin. In addition, the market value of these cryptocurrencies occupies most of the cryptocurrency capital market. It determines the development of the entire cryptocurrency market transaction, so we decided to divide these data into 3 groups according to mutual information of these cryptocurrencies market price in dollar with Bitcoin price. The first group is the Ethereum price, which has the largest mutual information with Bitcoin price. The second group is ETH and Cardano(ADA). The third group is the data collection of all cryptocurrencies. We can explore the impact of various cryptocurrencies'

prices on a single bitcoin market price through this operation.

Predicated on the results from the mutual information, the following blockchain transaction interactive features of Table 2.2 were finally determined as the dataset to train several machine learning models.

Table 2.2.: Cryptocurrency features used in Machine Learning models

Approach	Feature set
base	$Price, totaltx$
fl (Filtration)	$Price, totaltx, O^{\epsilon_1} \dots O^{\epsilon_S}$
betti	$Price, totaltx,$ $\beta_0(\epsilon_1), \dots, \beta_0(\epsilon_S), \beta_1(\epsilon_1), \dots, \beta_1(\epsilon_S)$
betti_der	$Price, totaltx,$ $\beta_0(\epsilon_1), \dots, \beta_0(\epsilon_S), \beta_1(\epsilon_1), \dots, \beta_1(\epsilon_S),$ $\beta_0'(\epsilon_1), \dots, \beta_0'(\epsilon_S), \beta_1'(\epsilon_1), \dots, \beta_1'(\epsilon_S)$

In the same way, Table 2.3 shows the arrangement of several Cryptocurrencies' prices dataset for the training of the machine learning models.

Table 2.3.: Cryptocurrencies' prices used in Machine Learning models

Approach	Feature set
base	$BTCprice, BTCtotaltx$
eth	$BTCprice, BTCtotaltx, ETHprice$
eth_ada	$BTCPrice, BTCtotaltx, ETHprice, ADAprice$
total	$BTCPrice, BTCtotaltx, ETHprice, ADAprice, LTCprice$ $DOGEprice, XRPprice, BN Bprice, USDTprice$

3. Methodology

Since the sudden expansion of the Bitcoin market price and the implementation of Blockchain technology in different fields, research around Bitcoin, cryptocurrency, and blockchain, has attracted innumerable scientific minds during the last few years. Among them, the unique attributes of cryptocurrency, such as decentralization, open-source, and anonymity, link the research of cryptocurrency with dynamic graphs.

3.1. Learning Graph Representations

3.1.1. ChainLet

The potential to observe and analyze all transactions on this dynamic graph is provided by the publicly available design of Bitcoin's distributed ledger. Cunert proposed the concept of the chainlet to assess the local topological feature of the cryptocurrency, and to investigate the relationship between the chainlet and the prediction of Bitcoin's price in different periods of time [ADGK18].

The original Bitcoin graph is comprised of various components, such as blocks, transactions, and addresses of transactions. Additionally, a Bitcoin transaction consists of an input address, output address, and a transferred Bitcoin amount. Prior research has focused on the different superficial features of Bitcoin or Blockchain, such as total transaction amount, market capitalization, previous price, and input/output degree, and fail to acknowledge the interactions between these features.

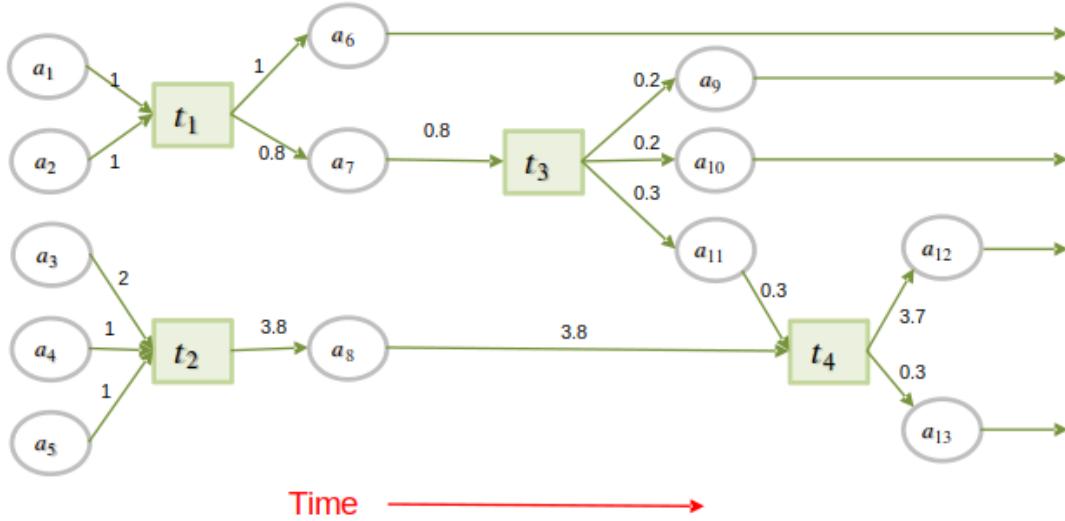


Figure 3.1.: The basic structure of a Bitcoin transaction.

In a chainlet, $C_{x \rightarrow y}$ signifies that it has x inputs and y outputs; connoted on the concept of the

3. Methodology

chainlet, three further concepts were established. The first concept of the chainlet is ‘merge’, in which the quantity of input branches is more than the output branches’ amount in a transaction. The next two concepts of chainlets are ‘transaction’ and ‘split’, which separately represent two chainlets that possess equal quantities of input and output, with the input amount being greater than the output amount [ADGK18].

Figure 3.2 articulates the visualization of the percentages of the chainlet in the Bitcoin Blockchain.

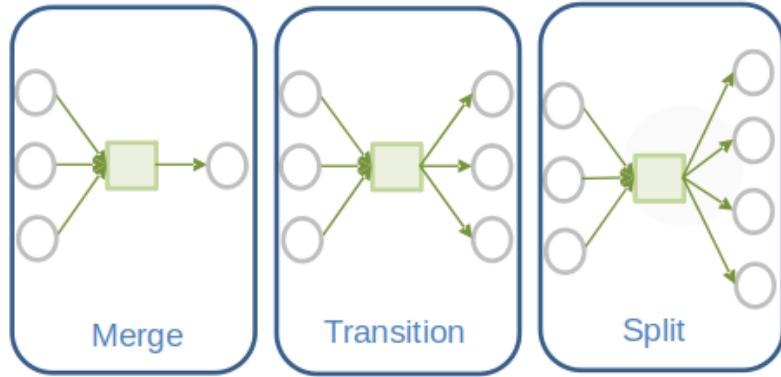


Figure 3.2.: Chainlets for 3 inputs: Merge ($C_{3 \rightarrow 1}$), Transition($C_{3 \rightarrow 3}$) and Split ($C_{3 \rightarrow 4}$)).

The chainlet approach structures the Bitcoin Blockchain graph with chainlet density by using the frequency of specific shapes. However, the chainlet suffers two disadvantages. Firstly, a complete chainlet may possess thousands of inputs and outputs, so the 1-chainlets can derive millions of specific certain shapes. Secondly, due to the cost, the chainlet approach has omitted most of the critical information [ADGK18].

3.1.2. ChainNet

Using the concept of chainlet, which is comprised of three different aspects: transaction, merge and split [ADGK18], Nazmiye proposed a new learning graph representation for Bitcoin Blockchain [AAG⁺19]. Bitcoin Blockchain possesses two types of nodes - transactions and addresses of the input/output transaction. The ChainNet utilizes address and transaction nodes to formulate a heterogeneous Blockchain graph[AAG⁺19]. This Bitcoin Blockchain is a dynamic graph, with the transaction node being unique and the address node having the potential to be continued and repeatedly utilized to extend the graph.

In ChainNet, the aforementioned shortcomings have been successfully addressed through the use of appropriate k for the k-chainlet and transferring the amounts of each transaction to ultimately structure a new blockchain graph. The two most significant concepts are ‘occurrence’ and ‘amount’ matrices. An occurrence matrix is established predicated on the input and output quantity of each everyday transaction; the length of this square matrix equates to the difference between the maximum and minimum quantity of in/output addresses. Despite this, the input and output amount may be over a thousand, meaning a threshold to restrict the boundary must be established, further reducing the size of the occurrence matrix and increasing the experiment’s computing efficiency. The value of the occurrence matrix is the summation of the counting appearing frequency of specific chainlets on the everyday blockchain graph. In contrast to the occurrence matrix, the

value of the amount matrix denotes the summation of the transferred Bitcoin of a specific shape [AAG⁺19].

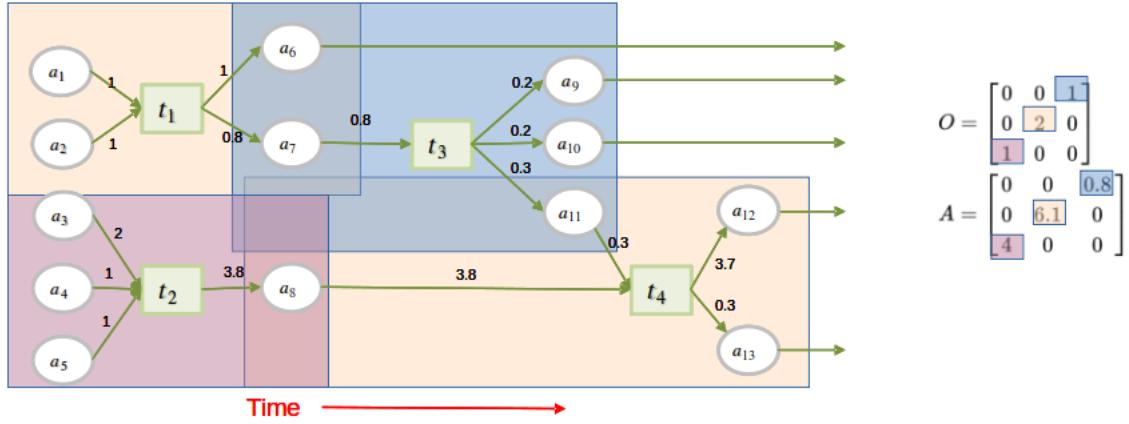


Figure 3.3.: A transaction-address graph representation of the Bitcoin network.

The rows and columns of the occurrence and amount matrices refer to the amount of input and output in the transaction, respectively. The difference between an occurrence and an amount matrix is that the value of the occurrence matrix is the sum of the number of transactions for a specific input and output. On the other hand, the value of the amount matrix equates to the sum of the transferred BTC values for a specific input and output.

Graph filtration: A user-defined occurrence threshold or number is set to filter the quantity of occurrence matrices. Through defining a set of scale values as boundaries, Graph Filtration stores the substructure of the Blockchain graph in order to train the prediction model. The following Graph Filtration algorithm represents the primary steps of processing the occurrence and amount matrix with scale values.

Algorithm 2 FL: Graph Filtration

```

Input:  $G$ : Blockchain graph, time  $t$ ,  $\epsilon_1, \dots, S$ : Set of  $S$  filtration scales. for  $\epsilon \in \epsilon_1, \dots, S$  do
     $O^\epsilon \leftarrow []$  ▷ initialize occurrence matrix
end
for  $Chainlet \mathbb{C}_{i \rightarrow j} \in G_t$  do
    for each scale  $\epsilon \in \epsilon_1, \dots, S$  do
        if  $\epsilon \leq amount(\mathbb{C}_{i \rightarrow j})$  then
             $O_{ij}^\epsilon \leftarrow 1 + O_{ij}^\epsilon$ 
        end
    end
end
return  $x_t = [O^{\epsilon_1}, \dots, O^{\epsilon_S}]$  ▷ concatenated Occurrence matrices
    
```

3.1.3. ETH-ChainNet

Due to the substantial difference between the Bitcoin and Ethereum Blockchain structures, in each Ethereum transaction of raw data, there are only single input and output addresses; the Ethereum

3. Methodology

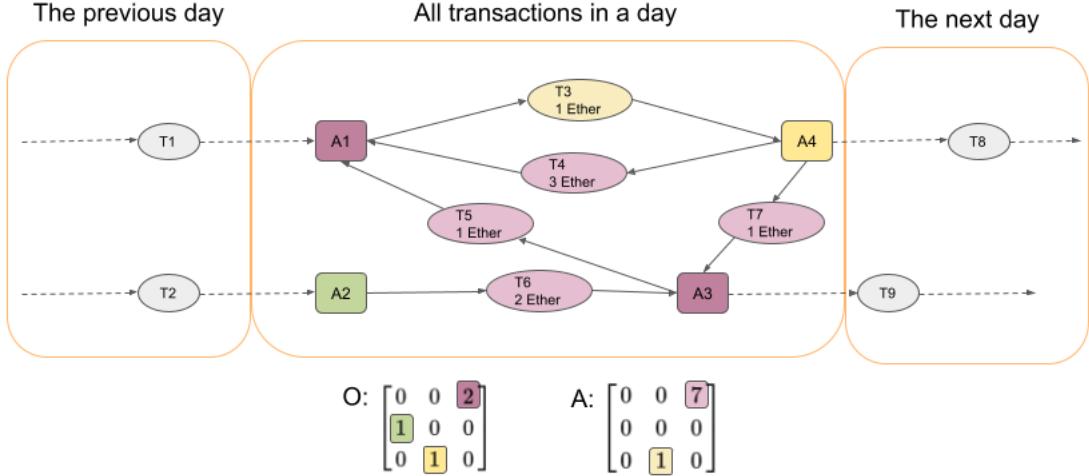


Figure 3.4.: A transaction-address graph representation of the Ethereum network

Blockchain dataset does not allow the previous ChainNet approaches to be implemented. Building off the ChainNet concept, a new approach was designed to organize the Ethereum occurrence and amount matrix, and subsequently employed this matrix to further calculate their Betti number, Betti derivatives, and train the machine learning models.

In the transaction-address graph representation of the Ethereum network exhibited in Figure 3.4, $\{A_1, \dots, A_4\}$ denotes different addresses, $\{T_1, \dots, T_7\}$ refers to different transactions in a day, the below matrix O represents the occurrence matrix, and A signifies the amount matrix. The rows and columns of the occurrence and amount matrices are similar to Bitcoin's network as it refers to the amount of input and output in the transaction, respectively. As some addresses do not have any amount of input or output in one day, the input and output start from 0. The value of the occurrence matrix is the sum of the number of transactions for specific input and output values. On the other hand, the value of the amount matrix is equal to the sum of the input Ether values for an address with a specific input and output.

3.1.4. Absorbing Random Walks (ARW)

The definition of the term ‘betweenness’ was first derived and proposed during research on shortest paths. Freeman calculated the fractional number of shortest paths between two arbitrary nodes and employed its fraction as the centrality measure in a graph, which is labelled as the ‘betweenness’ of the nodes [Fre77]. Subsequently in 2003, based on the assumption that information spread only along the shortest path, the method of using absorbing random walks to calculate betweenness was proposed by Newman. Newman’s measure of betweenness calculates the frequency of a node which is traversed between a pair of other nodes under the absorbing random walk [New05].

According to the statistics of the United Nations Office on Drugs and Crime, almost 2.1 trillion dollars or 3.6 percent of GDP is laundered in the US every year [UNO11]. Be that as it may, due to the complexity of money laundering, tracking the money from illicit activities is challenging for the government. Betweenness centrality is defined as a measure of the note of controlling information flowing between nodes, and its information flow is also the money flow of the transaction network, aiding us in identifying potential money laundering activities and patterns through exploring the

transaction network. Standard money laundry is comprised of three basic phases: placement, laying, and integration. During placement, the money is physically moved. Subsequently, in the layering phase, the money is propagated from the illegal source on the complex financial network to prevent the investigation from law enforcement authorities. During integration, under the cover of normal financial business, illegal money becomes legitimate business earnings [Reu05].

In legitimate business operation networks, the original betweenness approach is sufficient as money typically flows along the shortest paths. However, due to the nature of money laundering, more random paths are chosen rather than the shortest paths to cover the propagation of the illegal money. As a result, in the case of the cryptocurrency transaction, the absorbing random walks approach can efficiently implement the realistic betweenness centrality measure in which transactions flow along random paths.

The official definition of random walks was coined during the stochastic process, and refers to a path that includes a series of random walks on some mathematical space. To illustrate, on the integer number axis Z, when the random walk's start point is at 0, there is a 50 percent probability that the object moves -1 or +1 in the next step. Being a unique case of the random walk, the Markov chain denotes a stochastic process of random variables that satisfies the Markov property. In the concept of the Markov chain, the probabilities of the next step only involve the current state rather than the previous states. In the present research, the random walks are replaced by the Markov chain, with the transitions also replacing the status changes in the Markov chain. Furthermore, the transition probabilities represent the probabilities related to the various state changes in the transaction network.

1-ARW-betweenness centrality: Given a graph $G = (V, E)$ with $n = |V|$, and a restart probability α . The 1-ARW-betweenness centrality L_{arw} of a node $u \in V$ is defined as [Sch19]:

$$L_{arw}(u) = \frac{1}{n} \sum_{i=1}^n L_i$$

Figure 3.5 illustrates a small weighted transaction network. The weight of the network corresponds to the transaction value between two nodes, which is denoted by an adjacency matrix. The matrix located to the right is the matrix representation of the neighbor probability distribution N, in which the i^{th} input signifies the node of the i^{th} letter in the alphabet. Node A (row 1) has an output edge, causing ARW to move to node B with probability 1. Node B (line 2) possesses two output edges, D and G respectively [Sch19].

3.2. Learning Topological Representations

Research conducted in recent years has proved that the standard graph fails to capture transaction volumes and their potential relationship with the graph structure, as well as other important properties [GA15, Swa14]. Since there is a lack of basic approaches, the topological data analysis and persistent homology was proposed in order to capture these potential interactions of the graph structure.

3.2.1. Betti Number

For instance, $\mathbb{X} = \{X_1, \dots, X_n\}$ are a set of points in the Euclidean space, the ϵ_k is set as to scale according the adjacency matrix $A = \mathbb{1}_{d_{ij} \leq \epsilon_k}$ and then a graph G_k is established, with d_{ij} referring

3. Methodology

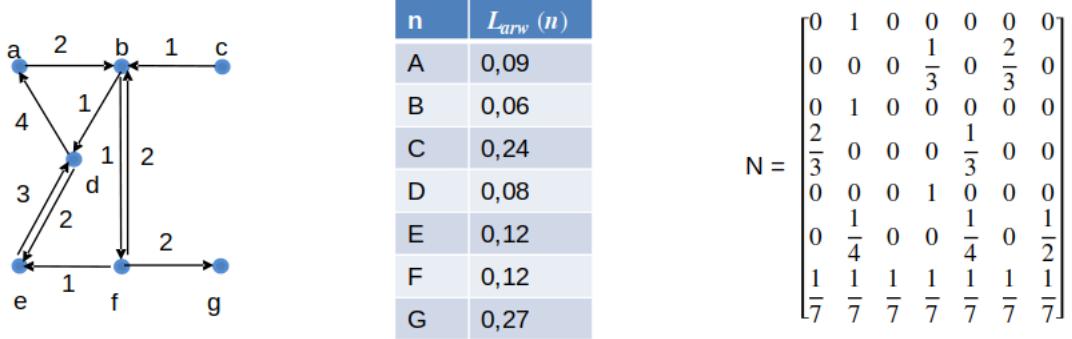


Figure 3.5.: The left graph is a visualization of the transaction network. Middle table refers the corresponding 1-ARW- betweenness centrality scores. To calculate the scores, the 1-ARW algorithm (Algorithm 1) is used, which will be explained further on. Right side is the matrix representation of the ARW neighbor-probability distributions. [Sch19]

to the Euclidean distance of X_i and X_j . Then, with the different scales $\epsilon_1 < \epsilon_2 < \dots < \epsilon_N$, a series of graphs is formed $G_i \subseteq G_2 \subseteq \dots \subseteq G_N$.

When the simplicial complex is related with graph collection $G_i \subseteq G_2 \subseteq \dots \subseteq G_N$, in order to collect the data of their intrinsic geometry, as the analogue of topological and combinatorial structure graphs, the topological features can be easily extracted and calculated. The use of multiple scales would be the best method for further reducing the process consumption of feature engineering and optimizing scale value. Due to the superiority of the structure and calculation, Vietoris-Rips has become a popular TDA in simplicial complex research.

Vietoris-Rips complex: VR_ϵ describes the Vietoris-Rips complex within the distinct value ϵ . As $k = 1, \dots, K$, the $(k-1)$ -simplices is the simplicial complex that has k elements of a set of data points $\mathbb{X} = \{X_1, \dots, X_n\}$, with the distance between their points being less than ϵ [HF96, Car09, Zom10]. The quantitative topological features can be counted through the VR filtration, $VR_1 \subseteq VR_2 \subseteq \dots \subseteq VR_N$.

In the present investigation, the betti number, betti sequence and betti derivatives are summarized to analyze the dynamical graph of the cryptocurrency. Subsequently, the Perseus software[Nan] is utilized to compute the persistent intervals and Betti numbers as its algorithm can efficiently complete this part of the experiment.

$$\beta_p = (\beta_p(\epsilon_1), \beta_p(\epsilon_2), \dots, \beta_p(\epsilon_N)), \quad p = 0, 1, \dots, K,$$

Here, the Betti number of p -simplices within the scale ϵ_N is denoted as $\beta_p(\epsilon_N)$

Betti numbers: In algebraic topology, the p -th Betti number signifies the rank of the p -th homology group, denoted as β_p , is the maximum cuts number to separate a surface into parts or 0, 1, ..., N -cycles. For instance, $\beta_0, \beta_1, \beta_2, \dots$ separately refers to the quantity of the connected components, loops and voids etc [AAG⁺19].

3.2.2. Betti Sequence

A benefit of utilizing the Betti sequences is their connection of nodes and the distance of the edges. Despite this, the complexity of their computation restricts its efficiency on the large transaction

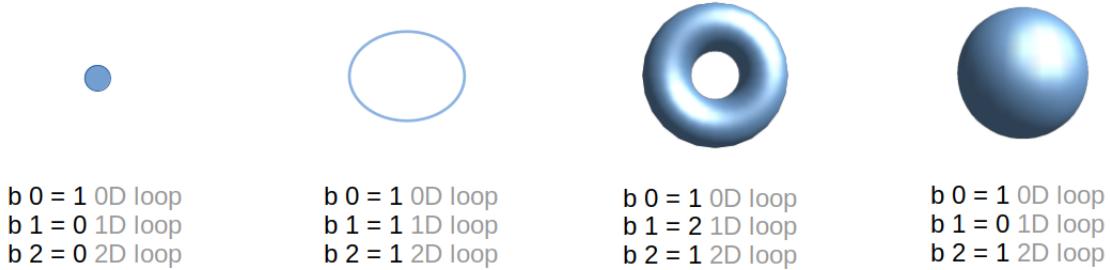


Figure 3.6.: Examples of Betti numbers in 0, 1, and 2 dimensional spaces.

network. In 2017, the Bitcoin Blockchain had extended 500K new nodes every day, and this number is still rapidly escalating at present, providing the most serious challenge of the Betti number application on the blockchain networks.

A potential solution to release the restrictions of the computational complexity of the Betti sequence is to form an N size of amount matrix - which is introduced in section 3.1.2. In the aforementioned matrix, each node represents a chainlet - which is introduced in section 3.1.1 - and the distance of each pair of chainlets is replaced by the edge distance of the pair of nodes in the amount matrix.

The following is the specific process of calculating the Betti number and Betti Sequence:

1. Convert Satoshis to Bitcoin:

$$a' = \log(1 + a \cdot 10^8)$$

In formula, a is an amount of Satoshis, a' is the log-transformed amounts of Bitcoin.

2. Calculate the k-th q-quantile $Q(k)$ of the chainlets:

$$\sum_{i=1}^{\tau} \mathbb{1}_{y_i < Q(k)} \approx \frac{\tau k}{q} \quad \text{and} \quad \sum_{i=1}^{\tau} \mathbb{1}_{y_i < Q(k)} \approx \frac{\tau(q-k)}{q}$$

where $k = 0, 1, \dots, q$, the equation to calculate the distance between node i and node j is:

$$d_{ij} = \sqrt{\sum_{k=0}^q [Q_i(k) - Q_j(k)]^2}$$

3. Establish a series of $\epsilon_1 < \epsilon_2 < \dots < \epsilon_N$ to filter the amount matrix of the whole year blockchain network, and produce a series of VR complexes $VR_1 \subseteq VR_2 \subseteq \dots \subseteq VR_N$ predicated on the 0-simplices and 1-simplices.

4. According to the above filtration of VR, the following formula is employed when computing the Betti Sequence of the Bitcoin Blockchain graph:

$$x_t = \beta_0(\epsilon_1), \dots, \beta_0(\epsilon_N); \beta_1(\epsilon_1), \dots, \beta_1(\epsilon_N);$$

This approach integrates the input and output degree and the transaction amount into an amount matrix. Through the combination of the distance and connectedness of edge, the quantitative information of the cryptocurrency graph is retained and the computation complexity on any large Blockchain network is decreased.

3. Methodology

3.2.3. Betti Derivatives

Since the p-th Betti curve and the Betti sequence are one in the same, evaluating the Betti curve through the dynamics of topological characteristics is another solution for analyzing the change rate of topological characteristics' influence on cryptocurrencies. The Betti derivatives aim is to achieve this goal:

$$\Delta^l \beta_p(\epsilon_k) = \Delta^{l-1} \beta_p(\epsilon_{k+1}) - \Delta^{l-1} \beta_p(\epsilon_k)$$

,

Where S refers to the filtration steps and k = 1,2,..,S-1, p = 0,1,..., these parameters are dependent on the situation. The change rate of the Betti curve is designed to systematically extract the dynamics of basic topological features and enhance the predictive ability. Built off the Betti derivatives, the Betti curve is integrated into the change rate derivatives, increasing the importance of the essential feature play in the final prediction. Since some essential features cannot be used to finish the filtration procedure, the accuracy of the experiment is notably reduced.

4. Machine Learning Models

Machine learning refers to the use of previous data sets to train the machine, so the trained machine can grasp the data's internal patterns and automatically solve problems relating to these data or similar data, such as data mining, natural language processing, and predictive analytics, etc [Dey16]. Due to the Internet Industry's recent boom, the research and applications of machine learning in various fields are rapidly developing. Traditional machine learning is comprised of supervised learning, unsupervised learning, and ensemble learning, etc [Dey16]. Juxtaposed to classic machine learning, the latest neural network approach has yielded excellent achievements in numerous fields including natural language processing, image processing, etc.

As the most efficient tools for predictive analytics, the Random Forest regression, Gaussian Process regression, ElasticNet, and XGBoost of machine learning models were employed to accept the Blockchain network and Bitcoin transaction. Although the classic CNN and RNN have evidenced that they do not yield positive effects on the cryptocurrency price prediction, the newest concept of the attention mechanism exhibited outstanding experimental results in several fields [AAG⁺19]. A derived model informer was utilized as the representative of the popular neural network in the machine learning area to articulate the impact of the neural network on the Blockchain network dataset.

4.1. Random Forest

In order to enhance the accuracy of ensemble learning, 'bagging' was proposed by Breiman in 1996 and 'random split selection' was coined by Dietterich in 1998. Bagging was established through the random selection of the sample in the training set to feed each tree model; random split selection chooses a random value among the best selections of split for each node [Bre01]. Random forest (RF) provides a means to plot the organism distributions and aid people in comprehending the motivations of potential present and future distributions. Due to RF's superior predictive capability, mapping potential future suitable habitats and presenting distributions are best suited to RF [PIL06].

Random forest is associated with the bagging (bootstrap aggregation) method in ensemble learning and is comprised of several decision trees, with no correlation between different decision trees in a random forest. Besides each decision tree is a method of machine learning which includes ID3, C4.5, and C5.0 generation algorithms; the decision tree is a tree structure algorithm. A judgment on an attribute is represented by each internal node, each branch denotes the output of a decision result, and finally, a classification result is signified by each leaf node. When the classification tasks are performed, new input samples are entered, and each decision tree in the forest is judged and classified separately, meaning each decision tree will possess its own classification result. Among the classification results of the decision tree, the random forest regards the tree with the most classification results as the conclusive result.

4. Machine Learning Models

4.1.1. Decision Tree

A decision tree can be defined as a tree structure algorithm and can be a binary or non-binary tree. Each none-leaf node denotes a test on a characteristic attribute, each branch refers to the output of this attribute in a certain value range, and each leaf node stores a category [FM99].

The procedure of the decision tree is as follows: it starts from the root node, then tests the corresponding feature attributes in the items to be classified, and subsequently selects the output branch according to its value until it reaches the leaf node. Finally, the category stored in the leaf node is utilized as the decision result [FM99].

The requisite components of a decision tree are:

1. Nodes and directed edges.
2. The two types of nodes: internal nodes and leaf nodes are present.
3. The internal node represents a feature, and the leaf node denotes a class.

Feature selection: Feature selection in the decision tree determines the features that are utilized when formulating a decision. In the training data set, there may be numerous features of each sample, and the effects of different features will differ. As a result, feature selection functions to filter out the features that are more relevant to the classification results, that is, the features possessing strong classification ability, so information gain is the criterion commonly employed in feature selection.

Decision tree generation: Following the feature selection phase, the information gain of all features are calculated for each node starting from the root node of the tree. Subsequently, the feature which provides the largest information gain in the decision tree is allocated as the split feature. Then, predicated on the different values of the split feature, the child nodes are established. The last step must be repeated until the information gain of each child node is minute or there are no features that can be selected.

Decision tree pruning: Because all features can be used to train the model in the concept of the decision tree, overfitting becomes the most frequent hindrance in the training of the decision tree. The pruning during the training was introduced as the most efficient approach against overfitting in the decision tree, actively removing parts of branches to reduce the risk of overfitting.

Depending on the decision, there are numerous derived popular algorithms, such as ID3, C4.5, and CART (Classification and Regression Tree). The oldest decision tree algorithm that was proposed is ID3; ID3 utilizes information gain to select features. Although C4.5 algorithm is an improved version of ID3, instead of directly using information gain, it introduces the "information gain ratio" indicator as the basis for feature selection. CART can be employed for classification or regression problems as the Gini coefficient is used instead of the information entropy model [FM99].

The accompanying benefits of decision trees are easy to comprehend and explain, and can be visualized and analyzed, and rules can be easily extracted. Decision trees can process nominal and numerical data simultaneously and is more suitable for processing samples with missing attributes. Additionally, it can handle irrelevant features when testing data sets; the run speed is relatively fast and it can produce feasible and effective results for large data sources in a relatively short time.

Despite the advantages of the decision tree, it suffers several shortcomings. For instance, overfitting occurs easily (random forest can greatly reduce overfitting), and the correlation of attributes in the data set are easily disregardable. Moreover, for those data with inconsistent sample sizes in each category within the decision tree, once the attributes are divided, different decision criteria will result in different attribute selection tendencies. Also, the information gain criterion favors attributes with a larger quantity of desirables (typically representing the ID3 algorithm), while the

gain rate criterion (CART) has a preference for attributes possessing a smaller number of desirabilities. However, when CART divides attributes, it no longer simply utilizes the gain rate to do so. Instead, it uses a heuristic rule (if the information gain is used, it suffers this disadvantage, similar to RF). When the ID3 algorithm calculates the information gain, the result is biased towards the feature with more numerical values.

4.1.2. Random Forest Regressor

The random forest can be employed not only for classification problems but also for regression. Differing from random forest classification, random forest regression predicts the weighted mean value of the target leaf node.

Figure 4.1 shows the overall picture of the random forest regression model. The sample input consists of massive data points, and the goal is to generate predictions for unknown values. The main steps of random forest regression are as follows:

1. X data points from the training set are randomly selected.
2. The decision tree associated with these X data points is constructed. Decision trees aren't favored as they use all data to construct trees, whilst random forest just selects X data in order to build trees.
3. The tree that will be constructed is selected and then steps 1 and 2 must be repeated.
4. The new input data is placed into each tree and its results are predicted separately. Finally, the new data point is assigned to find the average across all of the predicted values.

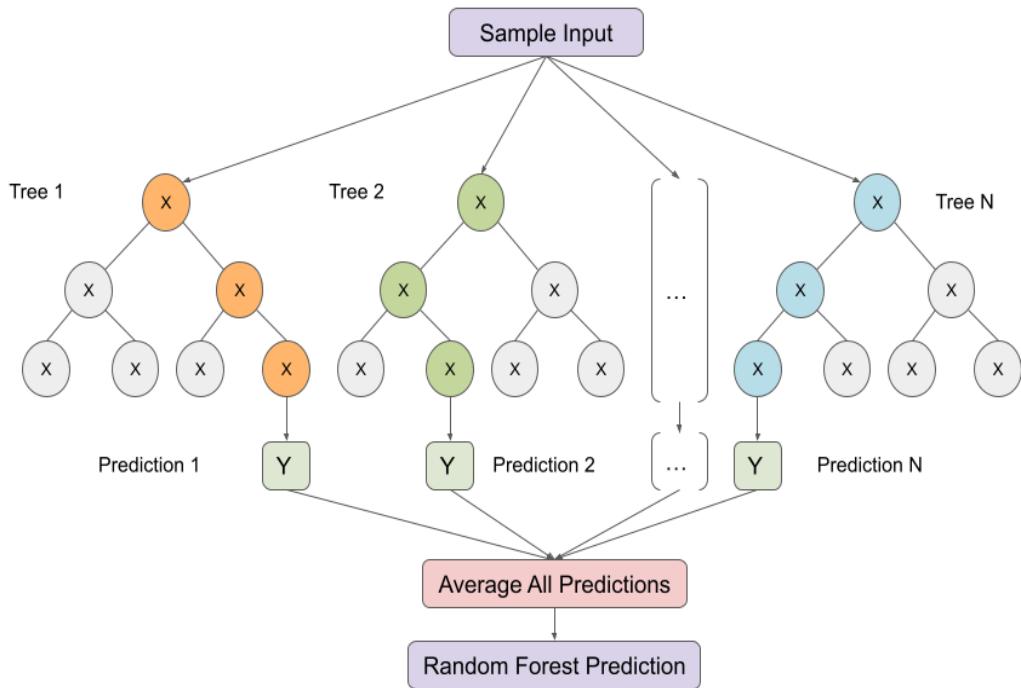


Figure 4.1.: An overall graph of the Random Forest model. [ZZP⁺20]

4. Machine Learning Models

Random forest regression suffers from drawbacks as it has been proven that overfitting causes noisy classifications and regression problems, and data with features that possess several different values cause features with more conditions to greater influence the random forest regression, so the weights of the features produced by the random forest on this form of data become unreliable. Despite this, random forest regression can process high-dimensional (many features) data without requiring dimensionality reduction or feature selection, and can judge the significance of features and the mutual influence between different features. Simultaneously, this algorithm improves the ability to resist overfitting and the implementation of the parallel method makes the training speed faster and easier. Regarding unbalanced data sets, this approach can balance bias, and despite the substantial part of the features that are missing, the excellent accuracy of the model can still be maintained.

4.2. Gaussian Process

An important component of traditional machine learning for data analysis and prediction is the supervised learning of the regression for the time series dataset. Due to the simple parameters caused by the lack of expressive power of the early models, Carl defined Gaussian processes and introduced them for regression in order to establish a Bayesian framework [Ras03, WR96]. The Gaussian process approach enhances the flexibility of machine learning for various practical scenarios.

Definition 1 A Gaussian Process is a collection of random variables, any finite number of which possess (consistent) joint Gaussian distributions [Ras03].

A Gaussian process is defined as a set of random variables, and any finite random variable from this set possesses a joint Gaussian distribution. In accordance with the definition of the Gaussian process, the mean and covariance functions are defined as $m(x)$ and $k(x, x')$, and the real process $f(x)$ conforms to the equations:

$$m(x) = E[f(x)],$$

$$k(x, x') = E[(f(x) - m(x))(f(x') - m(x'))],$$

$$f(x) \sim gp(m(x), k(x, x')),$$

,

In related research, there are numerous usual covariance functions that can be utilized to calculate the prior's covariance, such as:

Constant: $K_C(x, x') = C$,

Linear: $K_L(x, x') = x^T x'$,

White Gaussian noise: $K_{GN}(x, x') = \sigma^2 \delta_{x,x'}$,

Squared exponential: $K_{SE}(x, x') = \exp(-\frac{|d|^2}{2l^2})$,

Radial-basis function: $k(x_i, x_j) = \exp(-\frac{(x_i - x_j)^2}{2l^2})$,

Dot-Product: $k(x_i, x_j) = \sigma_0^2 + x_i \cdot x_j$,

$$\text{Matérn: } k(x_i, x_j) = \frac{1}{\Gamma(v)2^{v-1}} \left(\left(\frac{\sqrt{2v}}{l} d(x_i, x_j) \right)^v K_v \left(\frac{\sqrt{2v}}{l} d(x_i, x_j) \right) \right),$$

Concerning the implementation of the Gaussian process for regression, the specification of the mean prior to the Gaussian process is necessary. In the present experiment, the Gaussian Process Regression (GPR) from Scikit-learn was employed, and the prior mean was set as constant and zero by `normalize_y = False`. The prior covariance was controlled by the customized kernel object. From the grid search for the parameters of GPR, the DotProduct and WhiteKernel are the optimum selections to predict the price. Therewithal, the `n_restarts_optimizer` is specified as 2.

The biggest advantage of GPR is its probabilistic prediction that allows the user to adapt the fitting of the prediction in the interest region predicated on empirical confidence intervals. Furthermore, GPR provides several different kernels and a common kernel for different scenarios. Nonetheless, it suffers a low efficiency in over dozens of dimensional spaces, and their predictions must be calculated based on the whole samples or the features' information.

4.3. ElasticNet

One of the most employed algorithms in statistics is linear regression. Essentially, linear regression can express any mathematical relationship between two variables - output variables and predictor variables. Occasionally, this linear relationship exists between an output variable and several predictor variables. In regard to mathematical representation, a linear relationship is the simpler assumption to analyze data, and even many nonlinear problems can also be solved via linear models. As the simplest and most stable parameterization method, linear models are commonly employed in the industrialization scene.

In linear regression, the data is modelled by linear functions, and the unknown parameters in the model are estimated by utilizing existing data. For a multivariate linear regression model, it can be expressed as the following [ZH05]:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_p X_{ip} + \epsilon_i, i = 1, 2, \dots, n$$

Y serves as the linear function of X , and ϵ_i is bias. Linearity is the conditional mean of Y_i , which is linear in the vector parameter $\beta = (\beta_0, \beta_1, \dots, \beta_p)$. For linear regression:

$$y = b + \sum_{i=1}^n w_i x_i$$

In this formula, b is labelled as the bias, and w_i is the regression coefficient. Let $x_0 = 1$,

$$y = \sum_{i=0}^n w_i x_i$$

Through the calculation of the regression coefficient w_i , the linear regression model aims to obtain the linear regression model equation. The linear regression model refers to the proximity between the predicted value (prediction) and the label (actual value). The basic loss function of the linear regression model can be equivalent to the absolute loss function or squared loss function. The absolute loss function is [ZH05]:

4. Machine Learning Models

$$l = |y - \hat{y}|$$

y denotes the real value, \hat{y} is the predicted value, and $\hat{y} = \sum_{i=0}^n w_i x_i$. The squared loss function is:

$$l = (y - \hat{y})^2$$

Since the squared loss can be derived from every point on the model, the squared loss is typically employed as the loss function to adapt the linear regression model. When there are m training samples, and each sample possesses $n - 1$ features, the square loss function is derived as:

$$l = \frac{1}{2} \sum_{i=1}^m (y^i - \sum_{j=0}^{n-1} w_j x_j^i)^2$$

Precedingly, to minimize loss in the linear regression, the ordinary least squares (OLS) or Gradient descent algorithms predicated on the above loss function are utilized.

A popular optimization method in mathematics is the ordinary least squares, which attempts to establish either one or a set of estimated values as close to the actual value as possible. Ordinary least squares functions to predict unknown data through an existing dataset. For instance, in a two-dimensional coordinate, numerable points are scattered, so a straight line is drawn in an attempt to minimize the distance between these scattered points and the straight line. The essential purpose of the OLS is to determine the most probable function equation by minimizing the sum of the loss function. For example, there is a m training sample in the training set, and each training sample possesses $n-1$ features, meaning the prediction function can be expressed as:

$$Y = XW$$

Loss function:

$$(Y - XW)^T(Y - XW) = X^T(Y - XW)$$

Label Y is a $m \times 1$ matrix, the training feature X is a $m \times n$ matrix, and the regression coefficient W is a $n \times 1$ matrix. In the OLS, the derivative of W is:

$$\frac{d}{dW}(Y - XW)^T(Y - XW) = X^T(Y - XW)$$

Let the value of the first derivative be 0, then:

$$\widehat{W} = (X^T X)^{-1} X^T Y$$

Concerning the regression problem that accompanies more complex data, OLS's prediction accuracy is typically insufficient. Conversely, the correlation between the features may increase the model's complexity, and the strong correlation may result in a large variance of the results in the OLS model. This instability of the model can sometimes produce a sign that fails to match the actual meaning or cause of overfitting.

Regularization Ridge regression and lasso regression were first formulated to solve the overfitting of linear regression and the $(X^T X)$ irreversible problem during the process of calculating θ through the common equation method. Both ridge and lasso regression employ the loss function with regularization terms to achieve their aim. In the machine learning tasks, when the dataset's

features outnumber the sample points, the calculation of the $(X^T X)^{-1}$ can be hindered. The ridge regression model restricts the summation of θ_2 by using the penalty term λ , reducing the weight of the insignificant parameters. Like ridge regression, lasso regression also adds a regular term to limit the regression coefficient. This technique is known as *shrinkage* in statistics.

To prevent overfitting (θ is too large), a complexity penalty factor is added as a regular term after the objective function $J(\theta)$. The regular term can utilize L1-norm (Lasso), L2-norm (Ridge), or a combination of L1-norm and L2-norm (Elastic Net).

Lasso: L1-norm regularization

$$J(\theta) = \frac{1}{2} \sum_i^m (y^{(i)} - \theta^T x^{(i)})^2 + \lambda \sum_j^n |\theta_j|$$

Ridge: L2-norm regularization

$$J(\theta) = \frac{1}{2} \sum_i^m (y^{(i)} - \theta^T x^{(i)})^2 + \lambda \sum_j^n \theta_j^2$$

ElasticNet: L1-norm regularization + L2-norm regularization

$$J(\theta) = \frac{1}{2} \sum_i^m (y^{(i)} - \theta^T x^{(i)})^2 + \lambda (\rho \sum_j^n |\theta_j| + (1 - \rho) \sum_j^n \theta_j^2)$$

ElasticNet combines lasso regression and ridge regression into a single model with two penalty factors; the ElasticNet regression is proportional to the L1-norm and L2-norm simultaneously. Additionally, the model is as sparse as pure lasso regression and has the same regularization ability as ridge regression. ElasticNet's loss function is:

$$L(\bar{w}) = \frac{1}{2n} \| X\bar{w} - \bar{y} \|_2^2 + \alpha\beta \| \bar{w} \|_1 + \frac{\alpha(1 - \beta)}{2} \| \bar{w} \|_2^2$$

The ElasticNet of `sklearn.linear_model` package assisted in establishing the ElasticNet regression model, and according to the introduction of the `sklearn`, the parameter `l1_ratio` corresponds to β in the loss function while α corresponds to the α parameter of the loss function. `l1_ratio` = 1 signifies the lasso penalty but `l1_ratio` <= 0.01 is not reliable in the current ElasticNet regression model in `sklearn`. In the present experiment, the `alpha` = 10, and `l1_ratio` = 10 provided the best experimental prediction on the Blockchain Network.

4.4. XGBoost

XGBoost is a scalable end-to-end tree boosting system proposed by Tianqi in 2016 [CG16]. Due to its scalability in various scenarios including motion detection, web text classification as well as prediction for customer behavior and advertising click-through rate, and its high computing efficiency on a distributed system compared with other existing popular approaches, 17 of the 29 teams that won the 2015 competition chose this method [CG16]. In XGBoost, the essential components are the regularization, decision tree and random forest, with these concepts having already been introduced in subsection 4.1 and 4.3. Definition: combine the training loss and regularization in the objective function

4. Machine Learning Models

$$obj = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{i=1}^t \Omega(f_i)$$

In aiming to find the parameters of the tree, the additive strategy is used: at a time just add one new tree in the computation, then

$$\begin{aligned}\hat{y}_i^{(0)} &= 0 \\ \hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\ \hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\ \cdots \\ \hat{y}_i^{(t)} &= \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)\end{aligned}$$

At each step, add the tree that optimizes our objective:

$$\begin{aligned}obj^{(t)} &= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^t \Omega(f_i) \\ &= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + constant\end{aligned}$$

Here mean squared error (MSE) is used as loss function, then

$$\begin{aligned}obj^{(t)} &= \sum_{i=1}^n (y_i - (\hat{y}_i^{(t-1)} + f_t(x_i)))^2 + \sum_{i=1}^t \Omega(f_i) \\ &= \sum_{i=1}^n [2(\hat{y}_i^{(t-1)} - y_i)f_t(x_i) + f_t(x_i)^2] + \Omega(f_t) + constant\end{aligned}$$

Then the Taylor expansion of the loss function was took up to the second order:

$$obj^{(t)} = \sum_{i=1}^n [l(y_i, \hat{y}_i^{t-1}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + constant$$

Let g_i and h_i are

$$\begin{aligned}g_i &= \theta_{\hat{y}_i^{(t-1)}} \widehat{l(y_i, \hat{y}_i^{(t-1)})} \\ h_i &= \theta_{\hat{y}_i^{(t-1)}}^2 \widehat{l(y_i, \hat{y}_i^{(t-1)})}\end{aligned}$$

Then at step t the objective removes all constants

$$\sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)$$

Based on the aforementioned equation, every custom loss function can be optimized through the input g_i and h_T .

4.4.1. Model Complexity

Definition 2 *The tree $f(x)$ [CG16]*

$$f_t(x) = w_q(x), w \in R^T, q : R^d \rightarrow 1, 2, \dots, T.$$

Definition 3 *The complexity of the tree Ω as [CG16]*

$$\Omega(f) = +\frac{1}{2}\lambda \sum_{j=1}^T w_j^2$$

4.4.2. The Structure Score

Since all data points have the same score on the same leaf, the new definition of the complexity is used in the objective value:

$$\begin{aligned} obj(t) &\approx \sum_{i=1}^n [g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2] + \Gamma T + \frac{1}{2}\lambda \sum_{j=1}^T w_j^2 \\ obj(t) &= \sum_{j=1}^T [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2] + \Gamma T \end{aligned}$$

Let the data points indices set of the j -th leaf be $I_j = i | q(x_i) = j$, and define $G_j = \sum i \in I_j g_i$ and $H_j = \sum i \in I_j h_i$, then

$$obj(t) = \sum_{j=1}^T [G_j w_j + \frac{1}{2} (H_j + \lambda) w_j^2] + \Gamma T$$

Assemble the best objective reduction and a given structure $q(x)$, the best w_j is

$$\begin{aligned} w_j^* &= -\frac{G_j}{H_j + \lambda} \\ obj^* &= -\frac{1}{2} \sum_{j=1}^T \frac{G_j^2}{H_j + \lambda} + \Gamma T \end{aligned}$$

4.4.3. Learn Tree Structure

Because the of the large consumption in enumerating all possible trees picking the best tree, this solution optimizes one level each time, and splits a leaf into two leaves, so the gains function is

$$Gain = \frac{1}{2} [\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}] - \Gamma$$

The model splits the leaf which has the largest Gain until the largest Gain is less than zero, meaning that the error reduced by the split is not enough to make up for the loss caused by the complexity of the model, so a further split is unnecessary.

4.5. Latest Neural Network Model: Informer

In 2014, Kyunghyun evaluated the performance of the encoder-decoder approach on sentence translation and demonstrated that with the increase of sentence length, the performance of the neural machine translation was reduced [ZZP⁺20]. Despite the many attempts in previous research to use different architectures to form encoders, such as RNN and grConv, the curse of long-term memory remained.

However, the application of the attention mechanism in neural machine translation in 2017 was a turning point [VSP⁺17]. In the field of deep learning, this approach was considerably influential and groundbreaking. The attention mechanism proposed in the article is not only used in machine translation, but also extended to other application fields. Although attention mechanisms do not depend on any architecture, most attention mechanisms are currently attached to the Encoder-Decoder architecture.

4.5.1. Encoder-Decoder Architecture

The Encoder-Decoder architecture has existed for numerous years as a significantly popular model in the deep learning field. As an example, SegNet is a deep convolutional Encoder-Decoder Architecture in the application of Image Segmentation, and LSTM-LSTM is often used based on the Encoder-Decoder architecture in the neural network machine translation model [BKC17]. Machine translation is also the conversion of text to text and is called sequence to sequence learning, such as translating French into English [CVMBB14]. In the Encoder-Decoder architecture, the Encoder means to encode the input sequence into a fixed-length vector, and Decoder represents the process that decodes the previously generated fixed vector into the output sequence [CVMBB14].

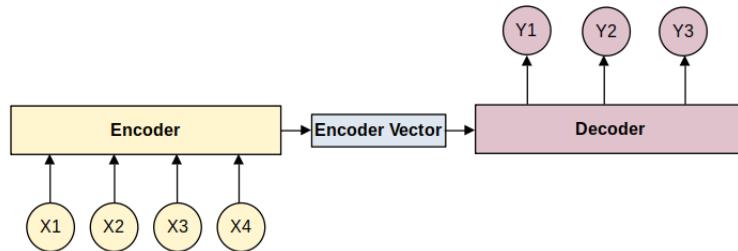


Figure 4.2.: Encoder-Decoder architecture

In the present research, the input is not the traditional sentence or paragraph of the text, but the sequence of the previous Windows-day features, which include totaltx, price, Betti number, betti sequence and filtration matrix. Subsequently, based on the Source of the input features, the Encoder-Decoder architecture can predict the price of cryptocurrency in the next day. The Source and Target comprise different features:

$$\text{Source} = \langle x_1, x_2, \dots, x_m \rangle$$

$$\text{Target} = \langle y_1, y_2, \dots, y_m \rangle$$

Then encode the input sequence of the features and transform the input sequences into an intermediate semantic representation C through non-linear transformation:

Transform the input sequence into an intermediate semantic representation context through non-linear transformation:

$$C = F(x_1, x_2, \dots, x_m)$$

For the Decoder, its task is to predict the price at time i

$$y_i = G(C, y_1, y_2, \dots, y_{i-1})$$

According to the intermediate semantic representation C of the Source and the historical information y_1, y_2, \dots, y_{i-1} that has previously been generated, When the horizon = 5 and window = 3, then the input sequence consists of features in the last 3 days and the y_i is the price sequence of the next 5 days.

When the Source is a Chinese sentence and the Target is an English sentence, the Encoder-Decoder architecture can solve machine translation problems. Similarly, this Encoder-Decoder architecture can be also used in projects involving text summaries, question answering systems and dialogue robots.

As well as being applied in the text field, the Encoder-Decoder architecture is also frequently used in speech recognition, image processing and other fields. By illustration, for speech recognition, the said architecture is completely applicable. The difference is that the input of the Encoder part is the speech stream, and the output is the corresponding text information; for the "image description" task, the input of the Encoder part is a picture, and the output of the Decoder is a description that can describe the semantic content of the picture. In general, the RNN model is usually adopted in the Encoder part of text processing and speech recognition, and the CNN model is usually adopted in the Encoder of image processing.

Despite the advancements brought about by the Encoder-Decoder architecture, it has a number of limitations. The biggest limitation is that the only connection between the encoder and the decoder is a fixed-length semantic vector Context C. To be specific, the encoder needs to compress the entire sequence of information into a fixed-length vector. There are two drawbacks to this. One is that the semantic vector Context C may not be able to completely represent the information of the entire sequence, and the other is that the information carried by the content input to the network will be overwritten by the information input later. The longer the input sequence, the more serious this phenomenon is. These two drawbacks render the decoder unable to obtain enough information from the input sequence at the beginning when decoding, and can lead to low accuracy of decoding.

4.5.2. Canonical Attention Mechanism

Bahdanau et al. proposed the attention mechanism in the paper "Neural Machine Translation by Jointly Learning to Align and Translate", so as to solve the two aforementioned drawbacks in the Encoder-Decoder architecture [BCB14]. In the field of deep learning, said paper was considerably influential and groundbreaking. The attention mechanism proposed in the article is not only used in machine translation, but has also been extended to other fields.

The inability to concentrate on key elements has been a persistent problem in the ordinary Encoder-Decoder architecture. For example, based on the aforementioned model:

$$Y_1 = f(C)$$

$$Y_2 = f(C, Y_1)$$

4. Machine Learning Models

$$Y_1 = f(C, Y_1, Y_2)$$

Where f is the nonlinear transformation function of the Decoder. When predicting the price on the target day, no matter which day's price is generated, the semantic Context C of the input sequence Source is the same. The semantic Context C is generated by Encoder encoding features in the Source, which means any feature in the Source has the same influence on the generation of a certain target prediction, regardless of y_1, y_2 or y_3 , illustrating the lack of the attention in the ordinary Encoder-Decoder architecture.

When the attention mechanism is applied in the above Encoder-Decoder architectures, the new model is similar to that shown in the following image.

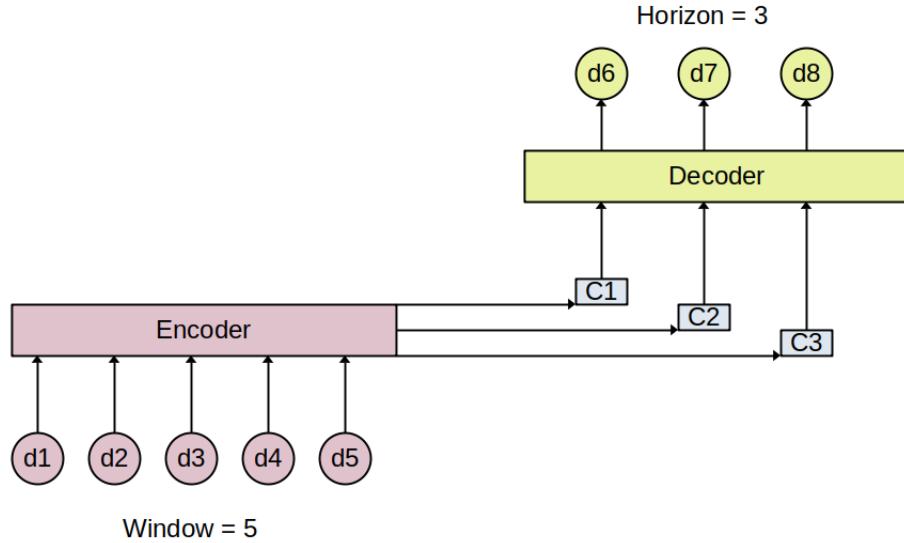


Figure 4.3.: Encoder-Decoder architecture with canonical attention mechanism

The process of generating the target prediction becomes the following form:

$$Y_1 = f(C_1)$$

$$Y_2 = f(C_2, Y_1)$$

$$Y_1 = f(C_3, Y_1, Y_2)$$

Each C_i corresponds to the attention probability distribution of different Source features. The Decoder aims to predict Y_i , and execute the some operations f_{att} between the previous output of the Decoder S_{i-1} and the output of the each hidden layer h_j , then normalize their result with SoftMax function to obtain the attention probability distribution, which is the attention weight a_{ij} of the model. Finally, the weighted sum of the input is calculated, and the expression Context C of the input sequence is calculated as the current partial input of the Decoder to generate Y_i .

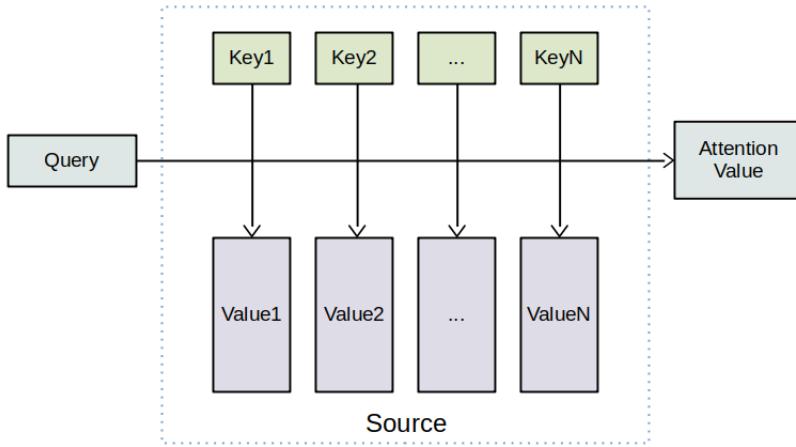


Figure 4.4.: The essence of Attention mechanism.

The Constituent elements in the Source are composed of a series of *< Key and Value >* data pairs. When a certain element *Query* in the Target is given, the similarity of correction between the *Query* and each *Key* is calculated to obtain the value corresponding to each *Key* weight coefficient, and then the *Value* is weighted and summed to obtain the final *Attention* value. Thus, in essence, the attention mechanism is used to perform a weighted summation of the *Value* in the Source, and *Query* and *Key* are used to calculate the weight coefficient of the corresponding *Value*.

Definition Attention Mechanism:

$$\text{Attention}(\text{Query}, \text{Source}) = \sum_{i=1}^{L_x} \text{Similarity}(\text{Query}, \text{Key}_i) * \text{Value}_i$$

$L_x = ||\text{Source}||$ is the length of the Source. The *Key* and *Value* in the Source point to the same thing. From a conceptual understanding, the attention mechanism is understood to effectively find out a small amount of important information from a large amount of data, and focus on these important features in the following process. The focusing process is reflected in the calculation of the weight coefficient, a larger weight of the corresponding *Value* means the information is more important.

Additionally, Source can be regarded as the content stored in the memory, and is composed of the address *Key* and *Value*. When there is a query of *Key = Query* of which the purpose is to retrieve the corresponding *Value* in the memory (*Attention* value), the query is addressed by comparing the similarity of the address of the *Query* and the element *Key* in the memory. However soft addressing differs from general addressing, wherein one record is found from the stored content, but the record may be retrieved from each *Key* address. The importance of extracting the content is determined based on the similarity of the *Query* and the *Key*, then the weighted sum of the *Value* is calculated as the *Attention* value.

The calculation of the *Attention* value can be summarized in three phases. The first phase is based on the *Query* and *Key* to calculate their similarity or correlation; the second stage is the normalization of the result from the first stage; and in the third phase, the weighted sum of the *Value* is calculated based on the weight coefficient of the second stage. The calculation process of the *Attention* value is abstracted into three stages as shown in the following Figure.

4. Machine Learning Models

In the first stage, different functions and computer mechanisms can be introduced. According to the Query and Key_i , their similarity or correlation is calculated, with the most popular methods including:

$$\text{Dotproduct : } \text{Similarity}(\text{Query}, \text{Key}_i) = \text{Query} \cdot \text{Key}_i$$

$$\text{Cosinesimilarity : } \text{Similarity}(\text{Query}, \text{Key}_i) = \frac{\text{Query} \cdot \text{Key}_i}{\|\text{Query}\| \cdot \|\text{Key}_i\|}$$

$$\text{MLPnetwork : } \text{Similarity}(\text{Query}, \text{Key}_i) = \text{MLP}(\text{Query}, \text{Key}_i)$$

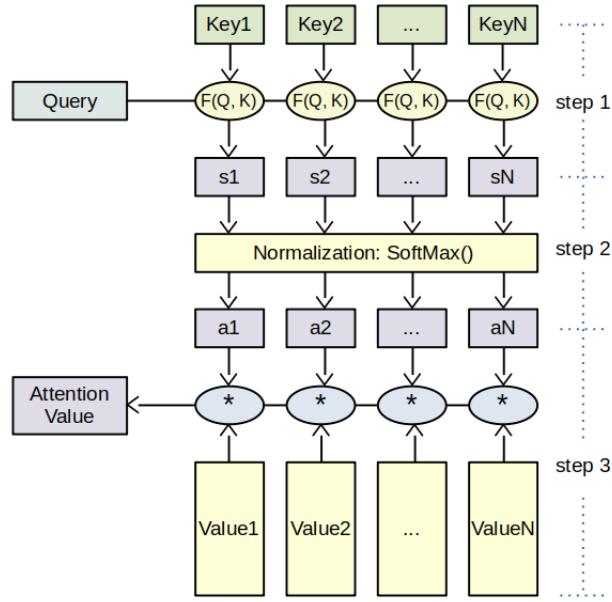


Figure 4.5.: Three-stage calculation Attention process.

The scores generated in the first stage have different value ranges depending on the different generation method. Thus, in the second stage, a calculation method such as SoftMax can be introduced to convert the scores in the first stage. On the one hand, this procedure can normalize the scores, and the original calculated score can be sorted into a probability distribution with the sum of all element weights being 1. On the other hand, the procedure can also highlight the weight of important elements through the internal mechanism of SoftMax.

$$a_i = \text{Softmax}(\text{Sim}_i) = \frac{e^{\text{Sim}_i}}{\sum_{j=1}^{L_x} e^{\text{Sim}_j}}$$

The result a_i of the second stage is the weight coefficient corresponding to $value_i$, and then weighted summation can be performed to obtain the Attention value:

$$\text{Attention}(\text{Query}, \text{Source}) = \sum_{i=1}^{L_x} a_i \cdot \text{Value}_i$$

4.5.3. ProbSparse Self-attention Mechanism

In 2017, Vaswani et al. proposed in their paper "Attention is all you need" that instead of using complex models such as RNN or CNN, relying only on the Attention model can parallelize training and allow for global information to be obtained [VSP⁺17]. Moreover, this self-attention model performs well on public data sets of different sizes in the field of machine translation.

The ordinary attention mechanism occurs in the similarity between the Source and Target, but the self-attention refers to the attention mechanism that occurs between the internal elements of the Source or the internal elements of the Target. During the calculation process, self-attention will directly connect any two elements in the Source through a calculation result, then the distance between the long-distance dependent elements is greatly shortened, which is conducive to the effective use of these elements. Through such approach, the long-distance interdependent elements can be more easily captured and the parallelism of calculation can be also enhanced.

Self attention: For the tuple input (query, key, value), perform the scaled dot-product [VSP⁺17]:

$$A(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V$$

where $Q \in \mathbb{R}^{L_Q \times d}$, $K \in \mathbb{R}^{L_K \times d}$, $V \in \mathbb{R}^{L_V \times d}$, and d is the dimension of input.

Several previous studies have shown that the distribution of self-attention probability is potentially sparse. In said studies, several "selective" counting strategies were designed for all $p(k_j|q_i)$ without significantly affecting performance, such as Sparse Transformer [CGRS19] and Longformer [BPC20], as well as LogSparse Transformer [LJX⁺19].

Query Sparsity Measurement: The probability $p(k_j|q_i)$ is defined as the i -th attention of the query on all keys. Due to the dot-product pairs of the attention mechanism, when the distribution of the $p(k_j|q_i)$ is close to uniform distribution $q(k_j|q_i) = \frac{1}{L_K}$, the self attention becomes small and redundant. Then, Kullback-Leibler divergence can be used $KL(q||p) = \ln \sum_{l=1}^{L_K} e^{\frac{q_l k_l^T}{\sqrt{d}}} - \frac{1}{L_K} \sum_{j=1}^{L_K} \frac{q_j k_j^T}{\sqrt{d}} - \ln L_K$. After the constant is dropped, the i -th query's sparsity measurement is [ZZP⁺20]:

$$M(q_i, K) = \ln \sum_{j=1}^{L_K} e^{\frac{q_i k_j^T}{\sqrt{d}}} - \frac{1}{L_K} \sum_{j=1}^{L_K} \frac{q_i k_j^T}{\sqrt{d}}$$

$\ln \sum_{j=1}^{L_K} e^{\frac{q_i k_j^T}{\sqrt{d}}}$ is the Log-Sum-Exp (LSE) of q_i on all the keys, $\frac{1}{L_K} \sum_{j=1}^{L_K} \frac{q_i k_j^T}{\sqrt{d}}$ is their arithmetic mean. The larger $M(q_i, K)$ of the i -th query refers to the more "diverse" attention probability p and the high chance of occurrence in the header field of the self-attention distribution.

ProbSparse Self-attention: According to the **Query Sparsity Measurement**, the ProbSparse Self-attention is defined as [ZZP⁺20]:

$$A(Q, K, V) = \text{Softmax}\left(\frac{\bar{Q}K^T}{\sqrt{d}}\right)V$$

,

\bar{Q} represents a sparse matrix of Q , and contains the Top- u queries based on the **Query Sparsity Measurement** $M(q, K)$. If we set a constant sampling factor c and $u = c \cdot \ln L_Q$, then the algorithm complexity of the dot-product is $O(\ln L_Q)$ and the memory usage maintains $O(L_K \ln L_Q)$.

4. Machine Learning Models

However, the calculation of the measurement $M(q_i, K)$ of all queries is quadratically $O(L_Q L_K)$, and the LSE is also potentially unstable. Thus, the bounds for each query and key are [ZZP⁺20]:

$$\ln L_K \leq M(q_i, K) \leq \max_j \left\{ \frac{q_i k_j^T}{\sqrt{d}} \right\} - \frac{1}{L_K} \sum_{j=1}^{L_K} \left\{ \frac{q_i k_j^T}{\sqrt{d}} \right\} + \ln L_K$$

then, the max-mean measurement is [ZZP⁺20]:

$$\bar{M}(q_i, K) = \max_j \left\{ \frac{q_i k_j^T}{\sqrt{d}} \right\} - \frac{1}{L_K} \sum_{j=1}^{L_K} \frac{q_i k_j^T}{\sqrt{d}}$$

The random sample $U = L_Q \ln L_Q$ dot-product pairs are already enough for $\bar{M}(q_i, K)$, and the other pairs are set as zero. With the sparse Top- u of these pairs as \bar{Q} , $L_Q = L_K = L$ causes the $O(L \ln L)$ time complexity and space complexity of the ProbSparse self-attention mechanism.

4.5.4. Self-attention Distilling

Due to the ProbSparse self-attention mechanism, the Encode stage generates redundant connections of Values [ZZP⁺20]. The distilling approach prunes the large time dimension of the input to focus the superior elements with dominating features and make a feature map of the focused self-attention in the next layer. The "distilling" procedure between j -th and $(j+1)$ -th layer is [ZZP⁺20, YKF17, GR17]:

$$X_{j+1}^t = \text{MaxPool}(\text{ELU}(\text{Conv1d}([X_j^t]_{AB})))$$

$[\cdot]_{AB}$ is comprised of the Multi-head ProbSparse self-attention mechanism and the basic attention operations, and the 1-D convolutional filters with kernel width=3 and $\text{ELU}(\cdot)$ activation function are used on the time dimension [ZZP⁺20]. Subsequently, a stride=2 max-pooling layer is used and the half sample after stacking a layer is obtained. This procedure leads to the memory usage $O((2 - \epsilon)L \log L)$, and the ϵ is considerably small. Further, the model will halve the replicas and drop one layer in a step by step manner to restrict the amount of the distilling layers. Finally, an aligned output dimension is provided and all outputs of the Encoder are combined as its final hidden representation [ZZP⁺20].

4.5.5. Informer Model

The Informer model is proposed by Haoyi in 2020, satisfy of the encoder-decoder architecture, and aim to solve the LSTF problem [ZZP⁺20]. Figure 4.6 is an overall graph of the Encoder-decoder architecture of the Informer model, at first massive long sequence inputs enter the Encoder, Encoder use proposed ProbSparse self-attention to replace the canonical self-attention to process the input dataset. After ProbSparse self-attention, the self-attention distilling operation is used to extract important attention, and sharpen the network size. In order to improve the robustness of the Encoder, the layer stacking replicas play an important role. For the Decoder component, the long sequence dataset as inputs enter into the decoder, replace the target data with zero, based on the concatenated feature map calculate the weighted attention composition, then generate the prediction for the replacement of the output dataset.

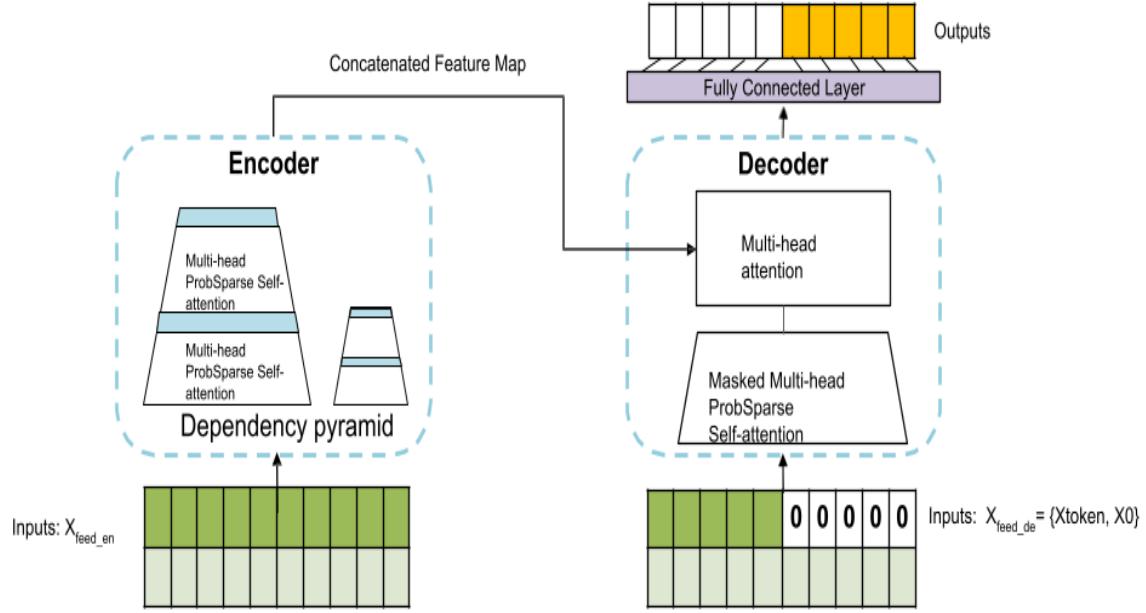


Figure 4.6.: An overall graph of the Encoder-decoder architecture of the Informer model. [ZZP⁺20]

According to the result of the previous experiment, the Informer model predicted excellent results than LSTnet based on CNN and LSTMa based on RNN and successfully decreased the MSE measurement [ZZP⁺20]. That experiments are performed on four datasets, including 2 collected actual datasets of **Electricity Transformer Temperature**(ETT) separated by time level as ETTh(1-hour) and ETTm(15-minutes), and 2 public benchmark datasets of **Electricity Consuming Load**(ECL) and the **Weather**.

5. Experiment

In this chapter the experimental parameters, actual processes and metrics of different predictive machine learning models are accurately explained based on the basic characteristics of cryptocurrencies, graph filtration, and their derivative Betti numbers, as well as other different interactive features. Subsequently, the evaluation methods which are applied to these various features and machine learning models are explained with different standards in two sections. In addition, the comparison of models in different periods and on different cryptocurrencies is introduced, such as Bitcoin and Ethereum.

5.1. Data pipeline

The second chapter has already introduced the basic information of Bitcoin, Ethereum, and other cryptocurrencies. This section focuses on implementing the data pipeline from the raw blockchain transaction data to the training dataset. Data pipeline consists primarily of 5 stages. The first step introduces how to gather and scrape raw data from the network. The Filtration and Aggregation are illustrated in the second part. The third subsection is the quantile calculation. Based on the Perseus software, the further application and explanation of the Perseus software were demonstrated in the fourth subsection. The last part is the influence of Principal Component Analysis with this different model and dataset.

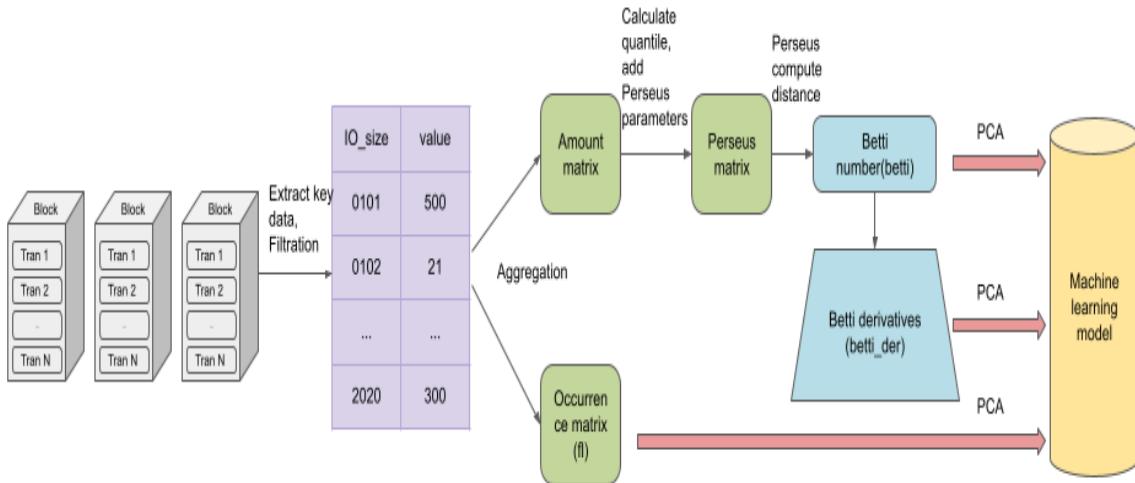


Figure 5.1.: The data pipeline overview diagram shows how raw data undergoes a sequence of processing into a training data set and enters the machine learning model.

5. Experiment

5.1.1. Gather Raw Data

For the Bitcoin blockchain dataset, we first choose the target period and convert the timestamp of this period to Unix timestamp. Through the API of [Blo] one by one, read the daily block raw data by the Unix timestamp. In each block consists of large related information. In this research, only the *vin_sz*, *vout_sz*, and *value* are extracted from the massive data and storage into a separate daily block data file, including the input and output address size of the transferred Bitcoin value of every single transaction in a day.

However, Ethereum has a different blockchain structure, and each Ethereum transaction has only one input and output address. Therefore the Ethereum data is formed by ETH-ChainNet (subsection 3.1.3). Since [Inf] does not provide service that read block by the Unix timestamp, general user can only search data by the block number or "latest". Our approach is trying to read the latest block, then used the latest block number to derivate the target block number range. The high-frequency query service of [Inf] allows the ordinary accounts to call 100000 times all API per day. But in 2020 the Ethereum network generated near to 3000000 blocks, using multi account and multi-process is a recommended measurement to reduce the time cost. The system receive the raw data of each Ethereum block, extracting the *from* and *to* address and the transferred Ether value and storing these primary information into a persistent separate daily block data file.

5.1.2. Filtration and Aggregation

Subsection 3.1.2 has explained the Filtration approach, follow this algorithm first set a threshold $\epsilon \subseteq \{5, 10, 20\}$ and based the ϵ to reform the data and generate a new variable *IO_SZ* which refers to the input and output size of a transaction. The aggregation of the same *IO_SZ* produces two matrices, the Amount matrix and the Occurrence matrix.

5.1.3. Quantile Calculation

Each value of the Amount matrix represents the node on the two-dimension surface, the process fellow the steps of calculating the Betti number and Betti Sequence (See 5.2.2). The *quantile* function of pandas is used to calculate the quantile, the effectiveness of *quantile_percentage* $\subseteq \{5, 20, 50, 100, 500, 1000\}$ are examined in the latter experiments, then the 500 is chosen as the best *quantile_percentage* to calculate the quantile for further exploration.

5.1.4. Perseus Application

According to the introduction of the Perseus software, the first row of the Perseus Matrix is the number of rows or columns in the symmetric distance matrix, and the second row represents the initial threshold distance g , step size s , a number of steps N and dimension cap C [Nan]. An example format of the input distance matrix is illustrated in Figure 5.2. Then we can apply Perseus software to this Perseus Matrix. In order to calculate the persistent homology for this input matrix, the following *distmat* command can be executed:

```
./perseus distmat (path to input distance matrix file) (output file)
```

However, the processing results of the Perseus software are not complete, and the Perseus software will automatically discard duplicate values in the output file. To fill in the missing content, the method introduced in Figure 5.3 is employed to generate the complete file:

perMat.json

3			
0.1	0.2	5	2
0	0.26	0.4	
0.26	0	2.1	
0.4	2.1	0	

3 : this is the number of rows/columns in the symmetric distance matrix
 0.1 0.2 5 2 : initial threshold distance g = 0.1, step size s = 0.2, number of steps N = 5 and dimension cap C = 2
 0 0.26 0.4 : distance from entry 1 to itself, entry 2, entry 3
 0.26 0 2.1 : distance from entry 2 to entry 1, itself and entry 3,
 0.4 2.1 0 : etc.

Figure 5.2.: An example format of the input distance matrix. the basic information is obtained from [Nan].

Perseus software output file

0	12	0
1	11	0
3	10	0
5	9	1
8	6	1

Fill in the missing content.



Complete file

0	12	0
1	11	0
2	11	0
3	10	0
4	10	0
5	9	1
6	9	1
7	9	1
8	6	1

Figure 5.3.: A demonstration shows how to yield the complete output file.

5.1.5. Principal Component Analysis

Jolliffe proposed principal component analysis (PCA) in 2005. It is a technique for reducing the dimensionality of prepared datasets, and it's not only increasing interpretability but also minimizing information loss of the dataset[Jol05]. PCA reaches this target by creating new uncorrelated variables to maximize variance successively. In this research, it is another major stage in the prepa-

5. Experiment

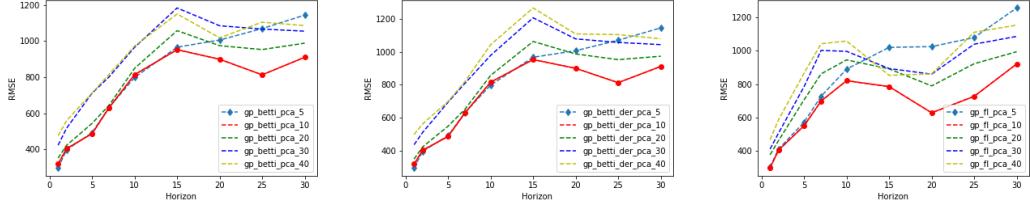


Figure 5.4.: When the window size is 3, the RMSE of the Gaussian Process changes with PCA = [5, 10, 20, 30, 40] based on the Betti number, Betti sequence, and Filtration dataset.

ration of the training data. For the training dataset of the vanilla machine learning model, the PCA is used to prune the weight of the 200 dimensional Betti number, Betti sequence, and 400-dimensional filtration matrix in the training dataset. In our experiment, the principal component analysis is implemented with the help of the *PCA* function of the *sklearn.decomposition* class in Python, and the *n_components* $\subseteq \{5, 10, 20, 30, 40\}$. Different *n_components* unexcepted cause huge differences in results for different types of data sets. Figure 5.4 illustrates the influence of PCA on the RMSE of the Gaussian process. In the three subgraphs, the line refers to the RMSE of the Gaussian process, and *gp_betti_pca_5* represents the prediction data generated by Gaussian Process regression on the Betti number data set processed by *n_components* = 10 of PCA. Comparing the data generated by different PCA, it can be seen that with the increase of *n_components*, the gap between the predicted value and the actual value gradually increases. On the other hand, when *n_components* is less than 10, its RMSE also increases because most of the imported information is lost with the unsuitable lower value of *n_components*. Therefore, in this experiment, using *n_components* = 10 as the best parameter, the dimensionality of the Betti number, Betti sequence and the filtered data set is reduced, and the information loss of dimensionality reduction is minimized at the same time.

5.2. Parameter Setting

For the traditional machine learning model, Grid search from Sklean is used to find the best parameters to optimize their RMSE. According to the hinweis of ChainNet, the number of trees is chosen from {10, 50, 100, 200, 300, 400, 500, 1000} for XGBT and RF models, and the learning rate is tried from {0.01, 0.1, 1.0}. L1, L2 and penalty constants of ENET are chosen from {0.0001, 0.001, 0.01, 0.1, 1.0, 10.0} and {0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0}. For the Gaussian Process, the Constant, Linear, White Gaussian noise, Squared exponential, and Dot-Product Matérn are used as the kernel options. Regarding the Informer model, the structure in Figure 5.5 illustrates the cooperation of the 4 convolutional layers, 2 MaxPool, and 3 Multi-heads attention in the Informer’s encoder. Each horizontal stack represents an individual encoder replica of the Informer, the upper stack takes the complete input sequence as the primary stack, the second stack uses half slices of the original data, then the self-attention mechanism is set in the red layer with dot product matrices. Distilling is applied on each layer to cascade decrease self-attention. Finally, the output result is the combination of the feature map of the two stacks.

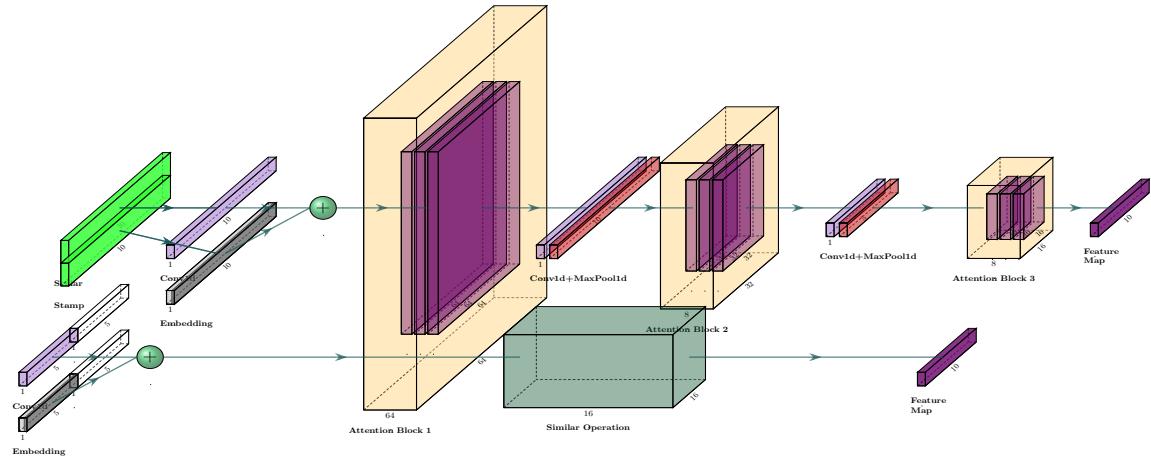


Figure 5.5.: The Informer model encoder architecture

5.3. RMSE of Models

First, the experimental result trained only on Price and the TotalTx dataset that is generated with a sliding window approach is set as the baseline of the machine learning model. As an example: when $w = 3$, the baseline model is trained on 6 features, like $[Price1, TotalTx1, Price2, TotalTx2, Price3, TotalTx3]$. In order to assess the model performance, the root mean square error(RMSE) is used.

Definition 4 *The root mean square error(RMSE)*

$$RMSE = \sqrt{\frac{1}{\|T\|} \sum_{t \in T} (y_t - \hat{y}_t)^2}$$

$\|T\|$ refers to the daily amount, and y_t and \hat{y}_t represent the actual price and predict price separated at the day t.

According to the practical experience of the ChainNet, the sliding window approach was chosen to train the model with 100 records, with every record consisting of a *window*-size training dataset and *horizon*-size actual data. Then, the prediction was performed once with the trained models. In the selection of the machine learning models and adjustment of the hyperparameters, ARIMAX [AAA14], XGBT, Random Forest, Gaussian Process regressor, ENET, and the neural network Informer model were chosen, and the GridSearchCV from *sklearn.model_selection* was used to tune the hyperparameter of the machine learning model.

Figure 5.6 shows the prediction RMSE result of the XGBT, Random Forest, Gaussian Process, ENET, and Informer models on the Bitcoin market dataset in 2017. Informer performance was higher than the other models only for $w = 7$ and $h \geq 25$, whereas Gaussian Process achieved the best RMSE values except for the former phase.

As a direct result of the economic stimulus policies of various governments in 2020, the cryptocurrency market exploded, and the transaction price of Bitcoin also rapidly increased. This excessive external stimulus has undoubtedly increased the instability and unpredictability of the market. In Figure 5.7, an observation can be made that compared with 2017, the RMSE value of each model

5. Experiment

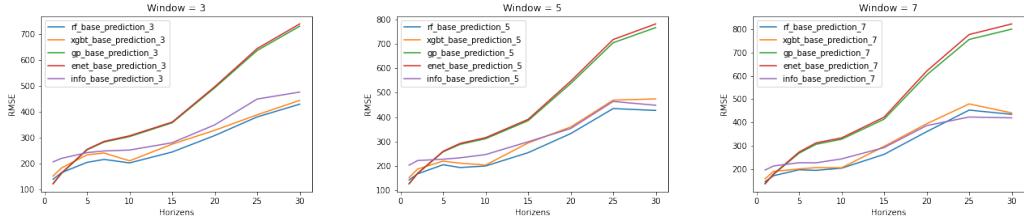


Figure 5.6.: RMSE of 5 models with window size 3, 5, 7 based predictions of 2017 Bitcoin prices.

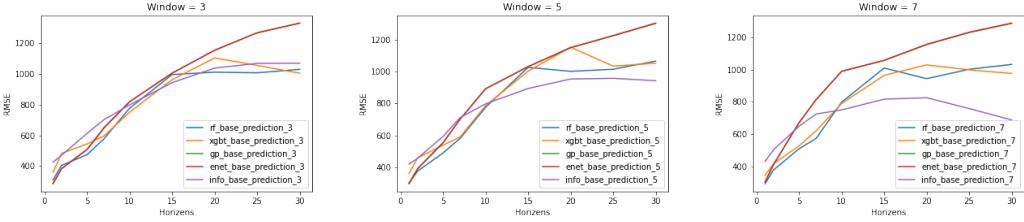


Figure 5.7.: RMSE of 5 models with window size 3, 5, 7 based predictions of 2020 Bitcoin prices.

has been expanded several times under the same parameters in 2020. However, the Informer model in second and third graph of Figure 5.7 shows a better convergence ability for strong noisy data, and when the *Horizons* is greater than 20, the RMSE of Informer for $w = 5$ and $w = 7$ begins to fall, until the prediction effect is significantly better than other traditional machine learning models.

Although, Nazmiye summarized the RMSE performance of ChainNet on the CNN and RNN models and demonstrated their shortcomings on the topological features on blockchain graphs [AAG⁺19]. However, the recent application of attention mechanisms in neural networks has enhanced the deep learning model on the time series prediction projects. On the other hand, the present experiment also proved neural networks have more advantages in the massive features dataset and complicate environment than the vanilla machine learning models, especially for the long-term prediction.

5.4. Performance of Topological Features

In this section, the performance of the machine learning models built with topological features Betti number and Betti sequence, as well as Filtration features is described. Recent research on Chainlet and ChainNet has already demonstrated the strength of these features in increasing model performance. In the present measurement for comparison between the performance of baseline and topological features, the percentage predictive gain was used as Performance Gain.

Definition 5 *Performance Gain*

$$\Delta_m(w, h) = 100 \times (1 - RMSE_m(w, h)/RMSE_{m_0}(w, h))$$

where m represents a specific machine learning model, and the baseline model is m_0 , then the baseline model m_0 and the comparison model m provides $RMSE_{m_0}(w, h)$ and $RMSE_m(w, h)$ respectively.

5.5. Performance of Other Cryptocurrencies

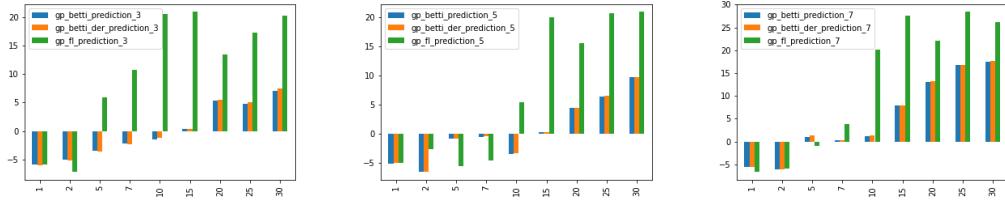


Figure 5.8.: Gaussian Process (GP) based regression performance in 2017.

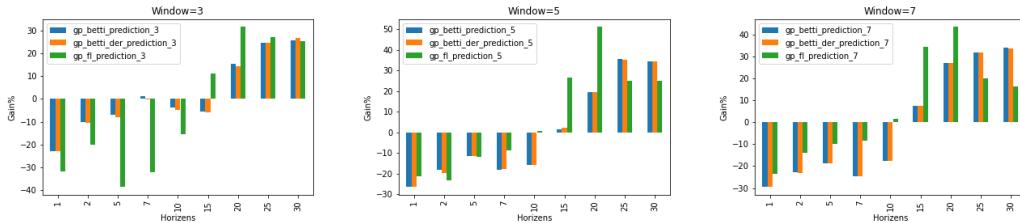


Figure 5.9.: Gaussian Process (GP) based regression performance in 2020.

The performance gain of the Gaussian Process (GP) regressor is shown in Figure 5.8, which indicates that the GP model trained with topological features was not better than the baseline performance in the first several days. In terms of the Random Forest and ENET results (In Figure A.4, A.5, A.8, and A.9) a similar trend can be seen. According to the result of the XGBT, the ChainNet features let the performance gain stay above 0%. However, the neural network model Informer shows decreasing negative gains on the topological features for increasing horizons, reaching a maximal negative gain of 150% for horizon 1. However, Figures A.10 and A.11 illustrate that FL enhances the predictive power of the Informer when the cryptocurrency price consistently increases in the long period. In this situation, Informer can capture this long-term change better than other traditional machine learning models, and assembling with the FL feature Informer model can further improve this capability. In contrast, the Betti number and Betti sequence do not have this ability.

The aforementioned results were achieved by PCA = 20 for ChainNet features. This setting stabilizes the best contribution of FL features to prediction, but there is still room for optimization for Betti numbers and Betti sequences. The results of the experiment indicate that the PCA has a different influence for all features on the blockchain graph.

5.5. Performance of Other Cryptocurrencies

In addition to the market transaction data of Bitcoin and the blockchain data of Bitcoin, we have also introduced transaction data of other cryptocurrencies which have a clear competitive relationship with Bitcoin in the capital market to explore the impact factors of Bitcoin price. According to the ranking of cryptocurrency market capital, [coia], the transaction data of Bitcoin, Ethereum, Litecoin, Dogecoin, Tether, Binance coin, Cardano, XRP are download from [Lim].

Section 2.4.4 introduces the structure of training data sets for other cryptocurrencies. In the RF and XGBT models, the result of the *total* cryptocurrency data set created the best enhancement for price prediction, reaching nearly 60 percent. There is no doubt that Figures 5.10 and Figure 5.13 also indicate the advantages of these two models for this type of data set, while the GP and ENET

5. Experiment

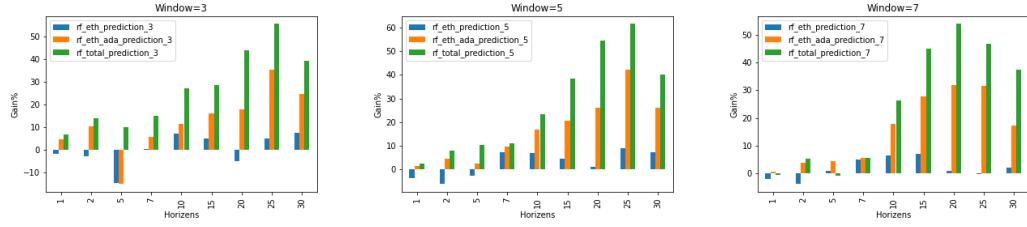


Figure 5.10.: Random Forest (RF) based regression performance in 2020.

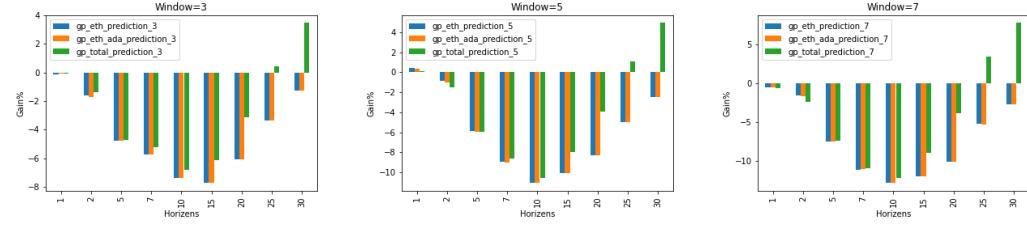


Figure 5.11.: Gaussian Process (GP) based regression performance in 2020.

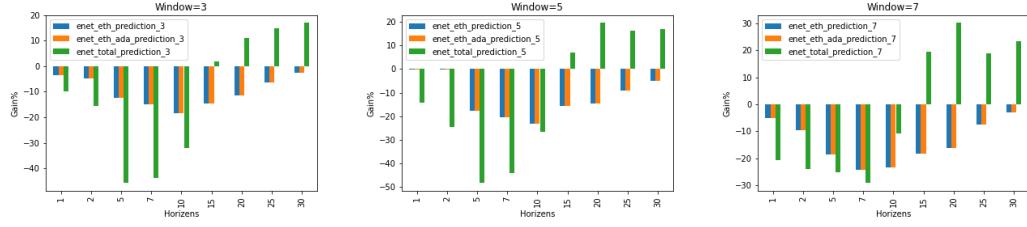


Figure 5.12.: Elastic net (ENET) based regression performance in 2020.

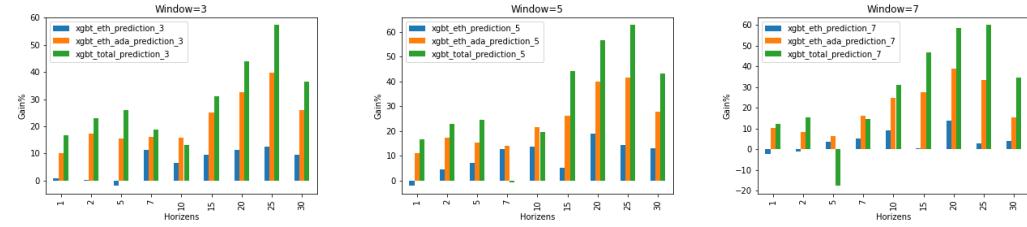


Figure 5.13.: XGBT based regression performance in 2020.

models are slightly inferior. In addition, although there is a high degree of mutual information between Bitcoin and Ethereum, the impact of Ethereum on the price of Bitcoin is even negligible.

5.6. Comparison on Different Blockchain

With similar approaches and machine learning models for Bitcoin, the impact of topological properties on the Ethereum dataset was examined, in addition to the performance of the Informer model on Ethereum. Although there is a big difference between the Bitcoin and Ethereum blockchain

5.6. Comparison on Different Blockchain

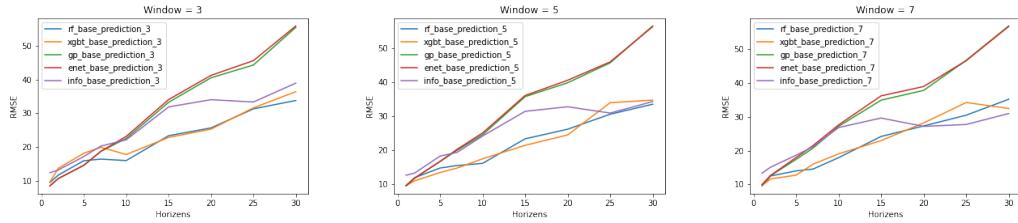


Figure 5.14.: RMSE of 5 models with window size 3,5,7 based predictions of 2020 Ethereum prices.

structures, and a different method was used to organize the Occurrence and Amount matrix on the Ethereum blockchain, the topological features display a similar impact on the Ethereum market price. Further, for horizons less than 20, the Random forest and XGBT model has predicted the closest result to the true value, and the RMSE of the Informer model is close thereto. However, the Informer model in Figure 5 and Figure 7 shows a better convergence ability, such as in the Bitcoin dataset for complex data, and when the *Horizons* is greater than 20, the RMSE of Informer for $w = 5$ and $w = 7$ begins to fall, until the prediction result is significantly better than other several models. All of the aforementioned observations demonstrate the impact of the topological features on the cryptocurrency blockchain for the cryptocurrency market price. Additionally, in the present research, the significant performance of the Informer model for long-term horizon cryptocurrency price prediction was also examined.

Tables 5.1 and 5.2 are the experimental records based on each model and various datasets of the Bitcoin blockchain. All the best RMSE in each horizon and window size concentrate on the topological features and all coins features. But these regulations are not clearly shown on Ethereum blockchain datasets. Consider the different data structure of the Bitcoin and Ethereum, the Occurrence Matrix of two cryptocurrencies are plotted in Figure 5.15. Based on the data structure of the Bitcoin matrix, the matrix has covered over 90 percent of the nodes. Still, the Occurrence matrix of Ethereum has obviously lost massive information. There are large of the node that has more 20 input and output degree. These lead to the ineffective of the topological features on the Ethereum dataset. However, the Informer model yields many remarkable predictions for the high horizon in Tables 5.3 and 5.4.

5. Experiment

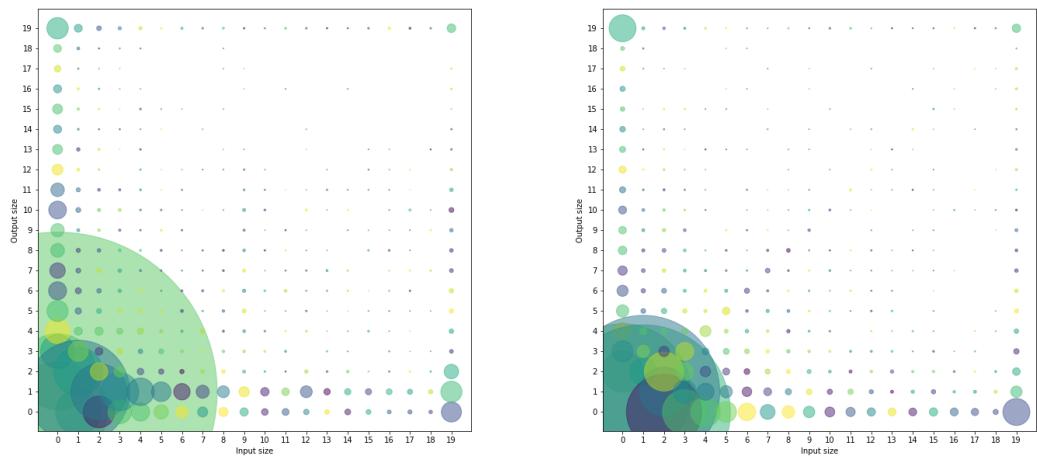


Figure 5.15.: The distribution of node on Bitcoin and Ethereum blockchain dataset, the data of left graph are from Bitcoin Occurrence matrix, the right's data are from Ethereum Occurrence matrix.

Table 5.1.: RMSE predicted based on the Bitcoin 2020 data set using various machine learning models, Window = 3

Model	feature	Horizon									
		1	2	5	7	10	15	20	25	30	
RF	base	311.27	404.79	474.51	577.93	775.79	994.15	1009.82	1005.60	1028.13	
	betti	301.14	375.92	553.50	551.69	757.02	801.77	802.63	721.03	807.08	
	betti_der	301.87	377.37	534.13	577.49	769.84	798.14	783.13	704.82	786.46	
	fl	278.70	366.42	480.14	603.74	767.61	868.66	889.61	945.59	977.59	
	eth	316.35	416.31	543.53	576.13	721.45	942.64	1058.83	953.80	950.92	
	eth_ada	296.88	363.22	545.81	544.53	686.81	834.13	830.66	651.59	773.57	
	allcoins	290.20	348.81	426.27	491.57	566.63	711.84	565.86	447.21	625.99	
GP	base	285.79	384.02	509.65	651.61	818.24	1003.51	1151.74	1265.01	1327.45	
	betti	351.55	422.64	544.14	644.46	849.10	1057.76	974.61	953.03	988.38	
	betti_der	350.93	424.25	549.64	652.88	856.76	1061.77	986.46	952.06	972.58	
	fl	376.03	461.56	705.68	861.17	944.97	891.15	788.98	922.07	994.06	
	eth	286.26	390.24	533.80	688.44	878.58	1081.03	1221.60	1307.47	1344.27	
	eth_ada	286.09	390.59	533.78	688.27	878.60	1081.05	1221	1307.48	1344.27	
	allcoins	286.11	389.23	533.53	685.17	873.96	1064.97	1187.91	1259.40	1281.57	
ENET	base	288.94	383.58	507.41	648.05	817.46	1003.44	1151.98	1265.45	1328.06	
	betti	353.43	425.43	544.28	643.30	848.64	1060.09	977.84	956.19	990.81	
	betti_der	352.59	427.51	551.00	651.81	856.79	1063.35	989.39	954.64	974.24	
	fl	384.15	471.10	716.08	868.19	950.71	898.84	790.70	925.10	994.23	
	eth	298.78	401.59	569.64	745.64	967.00	1150.02	1284.38	1347.78	1361.23	
	eth_ada	298.78	401.59	569.64	745.64	967.00	1150.02	1284.38	1347.78	1361.23	
	allcoins	317.34	442.99	739.22	931.50	1079.22	984.78	1023.48	1078.21	1103.52	
XGBT	base	361.28	482.65	543.88	596.40	749.89	960.92	1102.11	1053.55	1002.80	
	betti	354.29	430.75	639.27	665.98	717.97	844.37	829.44	735.67	768.59	
	betti_der	365.58	432.13	652.27	670.20	699.74	862.38	822.28	714.77	745.44	
	fl	400.91	468.99	600.26	620.45	706.63	902.08	861.10	826.19	933.46	
	eth	358.47	482.05	554.31	528.60	701.10	870.42	977.36	921.84	907.97	
	eth_ada	324.33	398.57	459.28	500.70	631.58	719.98	742.35	634.56	742.14	
	allcoins	301.19	371.91	402.79	484.03	651.80	662.68	616.45	449.75	635.97	
Info	base	425.88	470.64	612.74	701.84	797.25	941.82	1036.13	1066.90	1067.66	
	betti	1145.18	1016.87	1248.70	1122.61	1281.15	1223.44	1352.30	1451.81	1342.69	
	betti_der	1134.75	1167.65	1323.97	1315.58	1231.00	1357.84	1350.50	1330.50	1418.65	
	fl	379.48	389.77	422.36	414.71	529.76	578.97	666.59	672.57	647.60	

5. Experiment

Table 5.2.: RMSE predicted based on the Bitcoin 2020 data set using various machine learning models, Window = 7

Model	feature	Horizon									
		1	2	5	7	10	15	20	25	30	
RF	base	292.47	379.06	509.38	577.73	795.49	991.41	951.04	1001.79	1024.94	
	betti	291.24	366.93	454.08	501.99	584.59	617.33	542.02	668.79	727.58	
	betti_der	289.21	369.13	455.	504.76	586.72	616.04	559.28	668.26	738.32	
	fl	285.83	367.49	426.7	522.28	657.05	957.2	811.29	782.22	775.2	
	eth	299.28	393.13	505.04	544.14	745.36	937.09	935.04	1003.09	1010.6	
	eth_ada	291.86	363.95	486.56	540.72	654.12	729.18	643.33	685.1	853.86	
	allcoins	295.21	358.84	514.09	541.54	588.51	554.83	435.18	533.07	646.26	
GP	base	305.99	412.55	669.73	812.32	988.14	1055.79	1153.7	1228.28	1285.31	
	betti	396.21	506.75	794.83	1011.87	1162.1	975.16	840.61	837.02	848.18	
	betti_der	395.71	508.59	795.14	1012.3	1163.13	976.63	841.07	836.49	848.54	
	fl	378.36	470.02	736.74	881.36	972.43	691.34	650.32	981.51	1074.25	
	eth	306.09	418.91	720.47	903.36	1114.87	1182.3	1270.49	1293.1	1320.02	
	eth_ada	306.17	419.04	720.77	902.88	1114.96	1182.24	1270.5	1293.24	1319.68	
	allcoins	306.52	422.11	720.19	901.81	1109.05	1150.17	1198.42	1186.74	1185.83	
ENET	base	303.92	413.6	671.85	813.54	988.76	1056.79	1155.23	1230.27	1286.72	
	betti	401.19	514.25	797.68	1012.18	1163.6	977.58	845.64	842.74	850.63	
	betti_der	400.73	514.38	798.03	1012.7	1164.26	978.76	846.11	843.75	851.9	
	fl	383.59	477.24	740.56	884.78	980.95	696.44	654.6	980.85	1085.98	
	eth	319.72	452.66	797.16	1010.54	1218.52	1248.87	1342.62	1321.45	1324.62	
	eth_ada	319.72	452.66	797.16	1010.54	1218.52	1248.87	1342.62	1321.45	1324.62	
	allcoins	366.56	512.9	841.13	1049.88	1093.95	851.5	807.16	998.47	984.7	
XGBT	base	344.08	417.13	527.3	620.65	789.4	963.43	1028.34	997.73	975.19	
	betti	323.13	410.78	481.81	483.55	590.74	640.93	614.83	624.4	781.78	
	betti_der	322.79	409.97	487.46	475.61	592.21	644.35	627.83	622.01	773.73	
	fl	356.92	364.61	501.28	537.15	634.74	824.14	697.19	658.39	704.45	
	eth	352.12	422.	507.48	588.78	718.15	958.53	886.08	968.68	935.79	
	eth_ada	308.56	382.65	493.57	519.49	595.26	697.8	627.68	666.56	824.7	
	allcoins	302.4	353.61	620.3	530.27	545.2	513.88	427.49	400.88	639.14	
Info	base	431.33	504.01	648.93	724.55	749.9	816.13	825.35	758.41	687.63	
	betti	951.42	1137.94	1051.88	1077.18	1244.84	1169.59	1171.45	1281.15	1181.3	
	betti_der	1116.5	1095.1	1183.34	1188.45	1216.78	1265.9	1162.5	1284.48	1177.12	
	fl	348.09	335.52	436.93	506.92	538.93	606.72	581.09	506.21	599.63	

Table 5.3.: RMSE predicted based on the Ethereum 2020 data set using various machine learning models, Window = 3

Model	features	Horizon								
		1	2	5	7	10	15	20	25	30
RF	base	9.49	11.71	15.93	16.42	15.99	23.35	25.65	31.29	33.77
	betti	9.19	11.13	15.10	19.77	23.06	28.30	29.80	32.61	36.86
	betti_der	9.15	11.30	15.30	19.87	24.09	28.40	30.06	32.64	37.09
	fl	9.27	12.11	17.01	18.35	18.14	25.77	26.07	27.71	32.06
GP	base	8.49	10.65	14.55	18.70	22.54	33.24	40.49	44.27	55.42
	betti	13.17	17.63	24.88	30.96	30.30	37.93	42.34	47.39	51.90
	betti_der	13.10	17.85	24.45	30.34	30.24	38.14	42.37	46.76	52.02
	fl	13.81	19.81	25.23	33.38	33.89	30.96	28.70	29.43	28.39
ENET	base	8.52	10.68	14.62	18.70	23.15	34.04	41.20	45.56	55.83
	betti	12.33	15.81	23.20	29.82	29.67	37.29	41.75	46.06	50.44
	betti_der	12.11	16.31	23.44	29.53	29.43	37.35	41.78	45.79	50.83
	fl	14.11	18.05	24.62	32.66	32.86	29.52	28.89	28.90	27.73
XGBT	base	9.63	13.71	18.17	19.91	17.78	22.83	25.28	31.59	36.38
	betti	8.08	11.57	14.83	19.15	22.22	27.99	28.80	33.91	36.54
	betti_der	8.20	10.58	14.56	20.05	20.98	28.51	29.63	35.16	38.09
	fl	11.68	12.79	19.25	19.67	20.62	23.85	22.80	28.28	31.42
Info	base	12.38	13.22	17.18	20.27	22.13	31.87	34.01	33.32	38.91
	betti	15.29	20.31	20.87	25.02	25.49	35.72	38.10	43.15	45.25
	betti_der	15.84	15.78	20.62	21.65	26.31	35.65	37.06	41.29	46.97
	fl	728.50	825.10	754.36	795.22	638.12	638.50	661.63	983.70	762.38

5. Experiment

Table 5.4.: RMSE predicted based on the Ethereum 2020 data set using various machine learning models, Window = 7

Model	features	Horizon								
		1	2	5	7	10	15	20	25	30
RF	base	9.61	12.53	14.03	14.57	17.93	24.18	27.29	30.46	35.15
	betti	10.47	13.	14.88	15.82	21.84	27.57	30.32	32.5	39.42
	betti_der	10.24	12.88	14.84	15.95	21.77	27.24	30.14	32.66	39.26
	fl	9.83	13.73	17.15	17.48	19.46	24.11	25.8	26.4	35.53
GP	base	9.95	12.62	17.42	20.76	27.16	34.9	37.77	46.64	56.79
	betti	13.51	18.41	28.29	34.23	39.38	44.49	46.03	54.76	58.13
	betti_der	13.41	18.45	28.18	34.06	39.35	44.49	46.	55.39	58.84
	fl	15.52	20.64	26.79	29.79	29.71	33.58	31.52	31.96	40.38
ENET	base	10.01	12.69	17.95	21.52	27.59	36.17	38.92	46.53	56.64
	betti	13.32	18.35	28.34	34.04	38.84	44.27	45.42	53.78	56.76
	betti_der	13.36	18.41	28.39	33.86	38.53	44.35	45.47	53.49	56.86
	fl	16.16	19.41	26.08	28.3	27.66	32.28	31.56	28.7	37.66
XGBT	base	9.7	11.65	12.77	16.01	19.06	22.97	28.17	34.2	32.44
	betti	10.31	11.6	14.96	18.85	20.88	26.49	30.03	38.11	32.84
	betti_der	10.09	11.85	14.8	19.26	20.95	25.71	29.67	38.4	32.76
	fl	10.14	12.28	15.84	16.86	18.4	20.62	25.39	29.93	30.1
Info	base	13.37	15.13	18.65	21.18	26.81	29.62	27.15	27.71	30.95
	betti	15.64	22.34	21.18	20.94	26.33	32.55	29.54	36.14	41.74
	betti_der	17.44	20.15	19.78	21.51	26.49	29.55	31.9	32.42	36.73
	fl	674.35	643.8	656.54	730.94	808.65	850.03	746.75	826.2	781.99

6. Conclusion and Future Work

6.1. Summary and Conclusion

The topological properties of blockchain improve the cryptocurrency price prediction of Random Forest, Gaussian Process, XGBT, and ENET. However, the Betti number and Betti sequence have negative gains for the neural network with the Informer attention mechanism model. Informer reaches better performance in a situation wherein the price exhibits long-term increase. Informer assembly with FL features reaches the best performance gain in a chaotic situation. In addition, the results of the present research reveal that the topological features bring a maximal 40% prediction gain over the baseline model.

The factors that affect cryptocurrency are not only internal factors of blockchain, but also other cryptocurrency price changes during the same period. The price changes of Bitcoin, Ethereum, Litecoin and other cryptocurrencies show a strong correlation.

6.2. Future Work

Recent research in this field has focused on the internal factors of the blockchain, but if the dynamic graphs of all cryptocurrencies are assumed to be a huge network, the price of each cryptocurrency relates to the circulating capital in the huge network. The topological features can also be extracted from the entire cryptocurrency market and it's the interaction and impact on each cryptocurrency can be explored.

At the same time, the Informer model has room for improvement in time series forecasting projects. Future work can adjust the structure of Informer and optimize its hyperparameters to enhance its predictive ability on similar problems.

Finally, in the most relevant research, the window sliding method was used to train and predict the experimental target, but its efficiency is considerably low. The model must be trained many times before predicting and the process must be repeated numerous times to obtain the average RMSE. This method severely prolongs the running time. Thus, in the future, a better solution is needed to simplify this process and save actual running time.

A. appendix

A.1. Feature Engineering

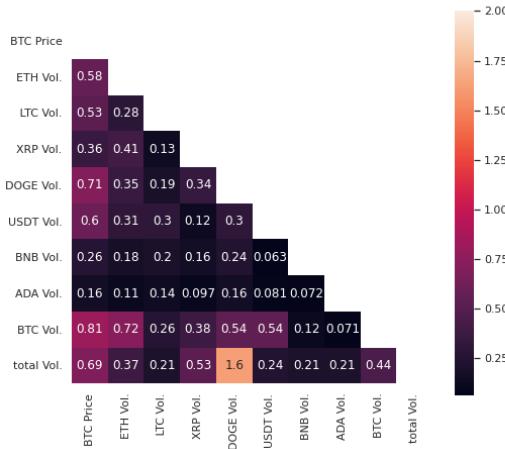


Figure A.1.: The covariance matrix of the transaction volume of the top 8 cryptocurrencies with the highest market capitalization.

The dataset of Figure A.1 is extracted from the [Lim]. The results in Figure A.1 show that there is a weak correlation between the transaction volumes of the top 8 cryptocurrencies. Only the transaction volume of Dogecoin and the total transaction volume of the top 8 cryptocurrencies have outstanding mutual information.

A.2. Bitcoin time series data

Since more than two nodes form a loop, it indicates that with the increase of ϵ , the simplices of a complex increase, the connected components decrease, and the number of loops increases, showing the negative correlation between Betti-0 and Betti-1. The Figure A.2 of the time series of daily log price, total transactions, average Betti-0 and Betti-1 numbers in 2020 have demonstrated this argument on the position of the red line, and Figure A.3 shows the opposite trend of the Betti-1 curve and the corresponding Betti-0 curve. Furthermore, we see in Figure A.3 that before the spike in Bitcoin in July, the Betti-0 was increasing, the Betti-1 and average daily transactions were decreased. In addition, extrema of the average daily transaction match corresponding spikes and plummets of Betti-0 and Betti-1 in July 2017.

A. appendix

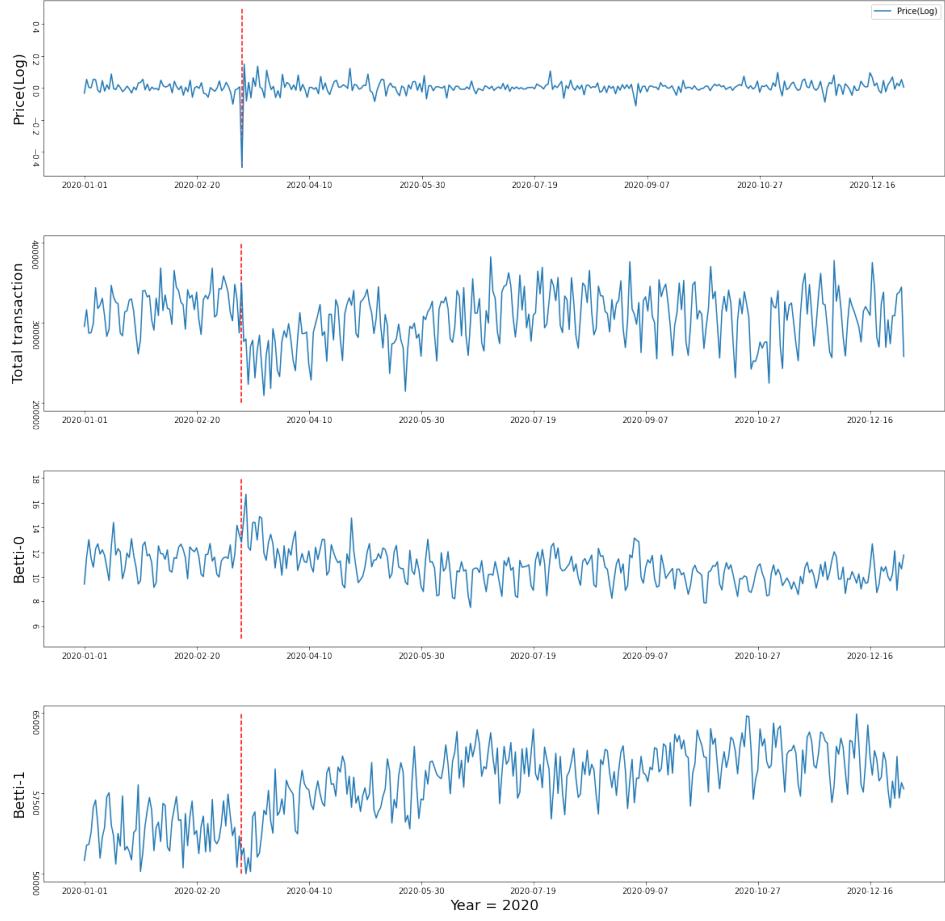


Figure A.2.: Time series of daily log price, total transactions, average Betti-0 and Betti-1 numbers in 2020.

A.3. Performance Gain of all models

This research primarily used seven kinds of machine learning models to study the interaction between the different properties and the price of cryptocurrencies. These models include ARIMA, Random Forest, XGBT, Gaussian Process, ElasticNet, RNN, and state-of-the-art of Informer. Due to the not converging results of the ARIMA and RNN, we persist only the results of the last five models. For the 2017 and 2020 Bitcoin blockchain datasets, we build a pipeline to process these two periods datasets. However, compared to the tiny fluctuations in Bitcoin prices based on the 2017 Bitcoin dataset, the rapid expansion of the cryptocurrency market in 2020 has significantly increased the difficulty of predicting cryptocurrency prices. In Figures A.5 and A.7, the apparent dramatic change of RMSE is expressed at the low horizon. In contrast, the ENET and Informer models display better stability in Figures A.9 and A.11; the dramatic change of the RMSE doesn't show in their process stage.

Although the Betti number and Betti derivative did not enhance the predictive ability of the Informer model for bitcoin prices, Informer more successfully absorbed the potential contribution

A.3. Performance Gain of all models

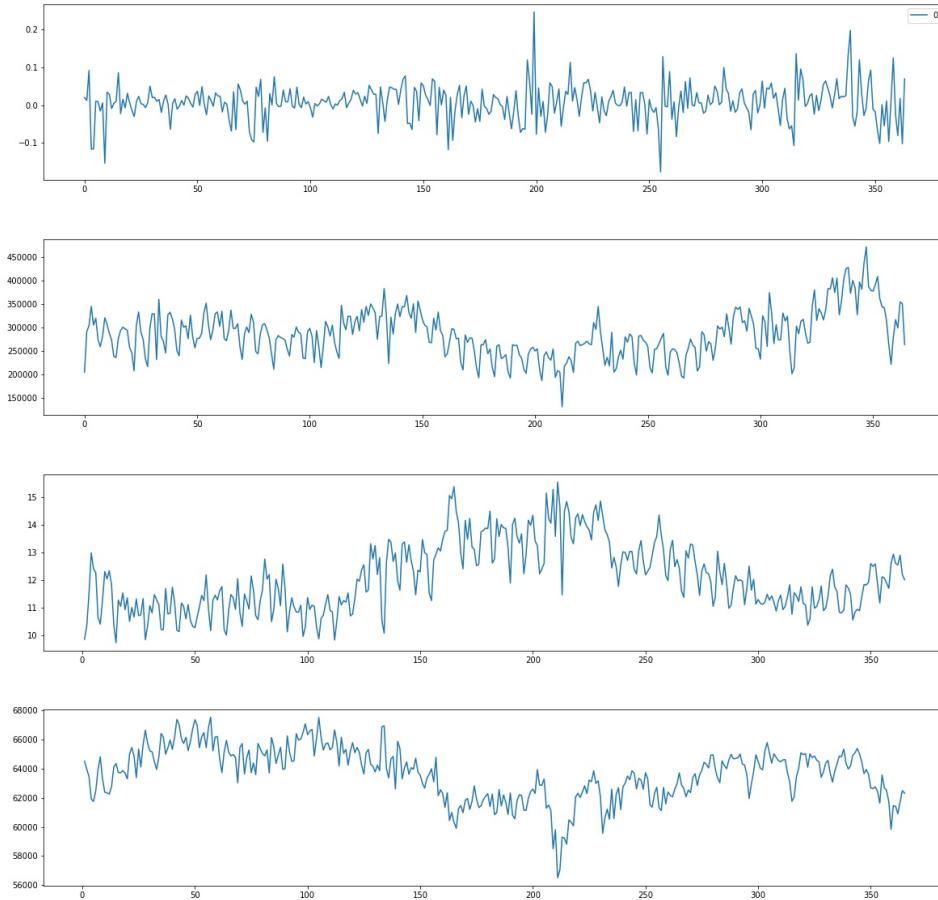


Figure A.3.: Time series of daily log price, total transactions, average Betti-0 and Betti-1 numbers in 2017.

of filtering features, leading to prominent and stable improvements in price prediction. Besides, in the experimental stage, we found that the Betti number, Betti derivative, and Filteration data sets have entirely different effects on prognosis in other PCA processes. When PCA = 10, the benefits of Betti number and Betti derivative for the machine learning model are more petite than PCA = 20, but the latter has more enhancement to fl.

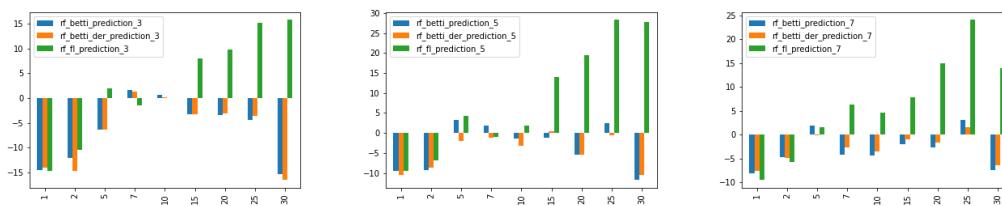


Figure A.4.: Random Forest (RF) model performance based on 2017 Bitcoin dataset.

A. appendix

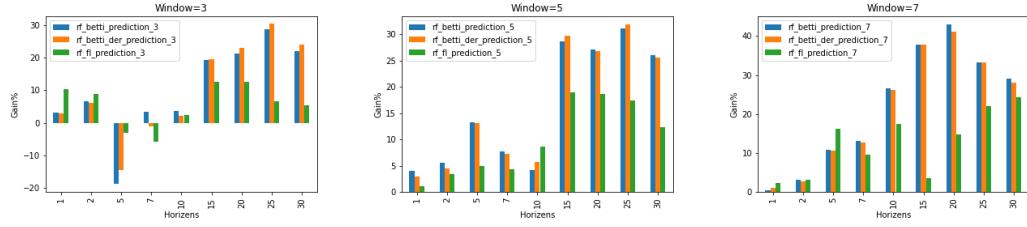


Figure A.5.: Random Forest (RF) model performance in 2020.

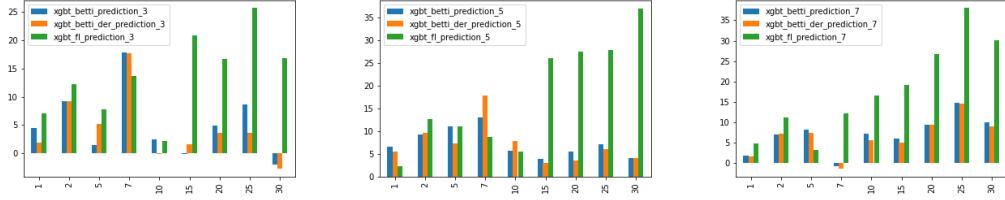


Figure A.6.: XGBT model performance based on 2017 Bitcoin dataset.

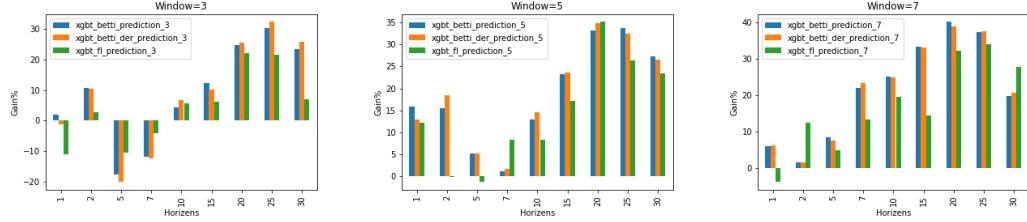


Figure A.7.: XGBT model performance based on 2020 Bitcoin dataset.

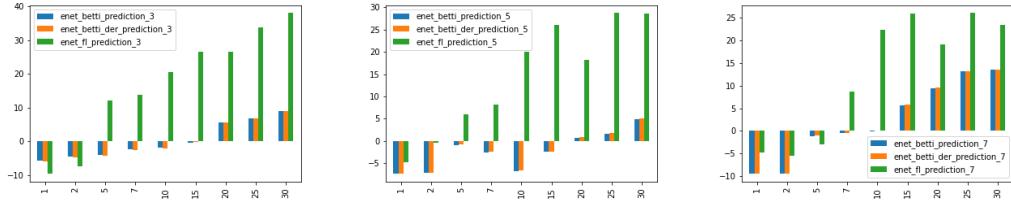


Figure A.8.: ENET model performance based on 2017 Bitcoin dataset.

A.4. Price prediction for 2016 with 1, 5, 10, 20 day horizons.

To more intuitively show the error between the predicted price of different dataset models and the actual market price, we use Figure A.12 - A.14 to plot the graph between the market price and different horizons over a period of time in 2016. Tabel A.1 is designed to organize the Bitcoin price, total transaction amount, and chainlet features in the training dataset based on the chainlet concept. *Pricelag1* refers to the first day's price before the current day, *#Translag1* represents the total transaction amount of the first day before the present, *SplitPattern* is the total number of daily transactions, which the amount of input address more remarkable than the number of the output

A.4. Price prediction for 2016 with 1, 5, 10, 20 day horizons.

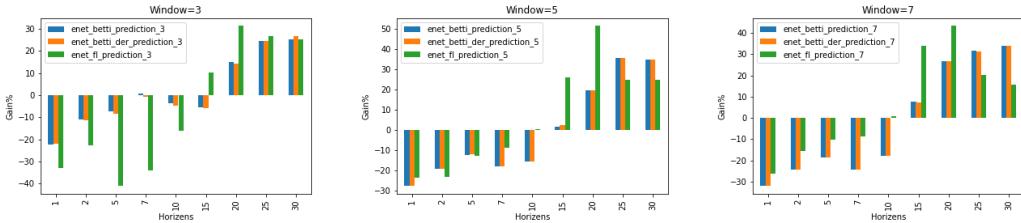


Figure A.9.: ENET model performance based on 2020 Bitcoin dataset.

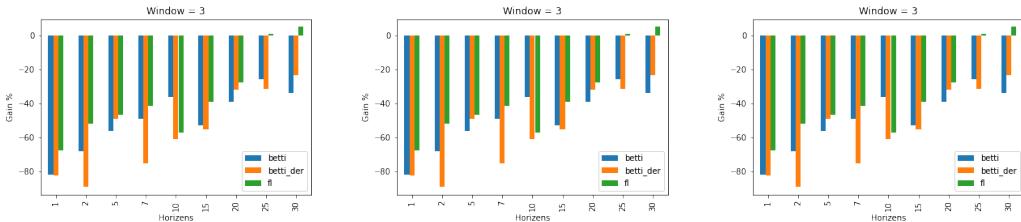


Figure A.10.: Informer based baseline performance based on 2017 Bitcoin dataset.

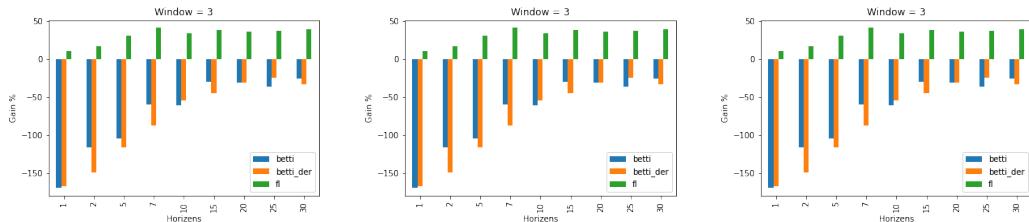


Figure A.11.: Informer based baseline performance based on 2020 Bitcoin dataset.

address, $C_{1 \rightarrow 7}$ is the total amount of the daily transaction which inputs amount equal one and output equal seven. Then we used actual price data as *Observed* line to observe the gap between the predictions of different models and the actual values.

In Figures A.12-A.14, we see that as the prediction range increases, the magnitude of interference to the predicted value also increases significantly, especially in the range where prices fluctuate sharply in June. This error is pronounced. On the other hand, the blue line represented by model 3, composed of chainlet elements, showed more vital anti-interference ability and created the best prediction record among all models. And baseline Model 0 demonstrates the worst results among all different range forecasts.

A. appendix

Table A.1.: Bitcoin Chainlet features used in Machine Learning models

Approach	Feature set
Model 0	Price lag 1, Price lag 2, Price lag 3
Model 1	Price lag 1, Price lag 2, Price lag 3, # Trans lag 1 , # Trans lag 2, # Trans lag 3
Model 2	Price lag 1, Price lag 2, Price lag 3, Split Pattern lag 1, Split Pattern lag 2, Split Pattern lag 3, Cluster 8 lag 1, Cluster 8 lag 2, Cluster 8 lag 3
Model 3	Price lag 1, Price lag 2, Price lag 3, $C_{1 \rightarrow 7}$ lag 1, $C_{1 \rightarrow 7}$ lag 2, $C_{1 \rightarrow 7}$ lag 3
Model 4	Price lag 1, Price lag 2, Price lag 3, $C_{1 \rightarrow 7}$ lag 1, $C_{1 \rightarrow 7}$ lag 2, $C_{1 \rightarrow 7}$ lag 3, $C_{6 \rightarrow 1}$ lag 1, $C_{6 \rightarrow 1}$ lag 2, $C_{6 \rightarrow 1}$ lag 3
Model 5	Price lag 1, Price lag 2, Price lag 3, $C_{1 \rightarrow 7}$ lag 1, $C_{1 \rightarrow 7}$ lag 2, $C_{1 \rightarrow 7}$ lag 3, $C_{6 \rightarrow 1}$ lag 1, $C_{6 \rightarrow 1}$ lag 2, $C_{6 \rightarrow 1}$ lag 3, $C_{3 \rightarrow 3}$ lag 1, $C_{3 \rightarrow 3}$ lag 2, $C_{3 \rightarrow 3}$ lag 3
Observed	Current price

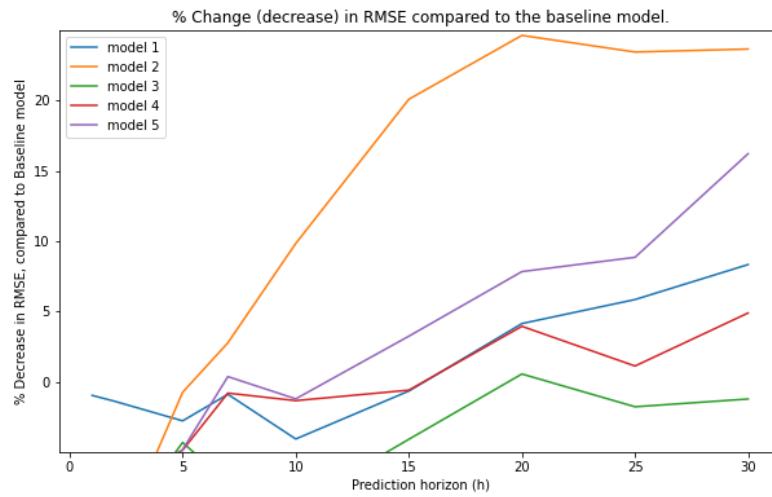


Figure A.12.: % Change (decrease) in RMSE compared to the baseline model.

A.4. Price prediction for 2016 with 1, 5, 10, 20 day horizons.

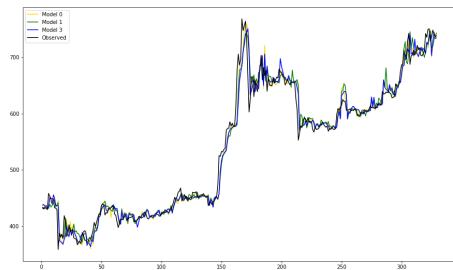


Figure A.13.: Price prediction for 2016 with 1 day horizon.

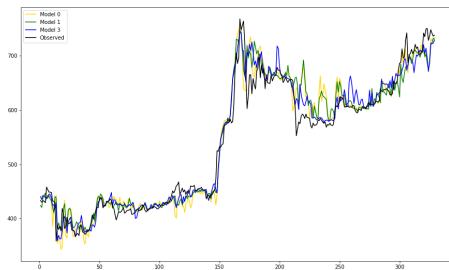


Figure A.14.: Price prediction for 2016 with 5 day horizon.

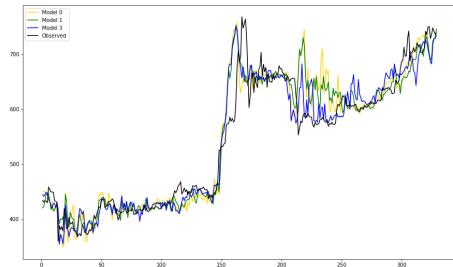


Figure A.15.: Price prediction for 2016 with 10 day horizon.

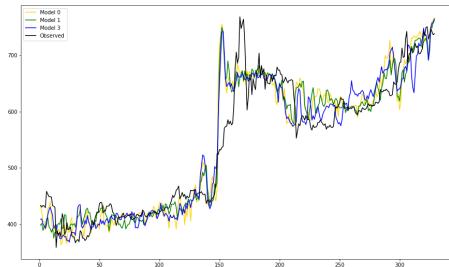


Figure A.16.: Price prediction for 2016 with 20 day horizon.

List of Figures

1.1.	An example of a blockchain composed of the genesis block and blocks A, B and other blocks. The system generates a new block every 10 minutes.	2
1.2.	The ranking of top cryptocurrencies Bitcoin and Ethereum in the world capital market.	3
2.1.	An example for BTC raw block data from blockchain.com.	6
2.2.	An example for BTC transaction data from blockchain.com.	6
2.3.	An example for ETH raw block data from Web3 API.	7
2.4.	An example for ETH transaction data from Web3 API.	7
2.5.	Bitcoin and Ethereum Market Price(2016-2021).	8
2.6.	XRP and Litecoin Market Price(2016-2021).	8
2.7.	Dogecoin Market Price(2017-2021).	8
2.8.	Price per day of Bitcoin in 2020.	10
2.9.	Total transaction per day of Bitcoin blockchain in 2020.	10
2.10.	Betti0 of Bitcoin blockchain in 2020.	10
2.11.	Betti1 of Bitcoin blockchain in 2020.	10
2.12.	Visualization of a sliding window, with window size w and h the prediction horizons.	12
2.13.	The covariance matrix of Bitcoin blockchain various properties.	13
2.14.	The covariance matrix of the top 8 cryptocurrencies price by market capitalization.	14
3.1.	The basic structure of a Bitcoin transaction.	17
3.2.	Chainlets for 3 inputs: Merge ($C_{3 \rightarrow 1}$), Transition($C_{3 \rightarrow 3}$) and Split ($C_{3 \rightarrow 4}$).	18
3.3.	A transaction-address graph representation of the Bitcoin network.	19
3.4.	A transaction-address graph representation of the Ethereum network	20
3.5.	The left graph is a visualization of the transaction network. Middle table refers the corresponding 1-ARW- betweenness centrality scores. To calculate the scores, the 1-ARW algorithm (Algorithm 1) is used, which will be explained further on. Right side is the matrix representation of the ARW neighbor-probability distributions. [Sch19]	22
3.6.	Examples of Betti numbers in 0, 1, and 2 dimensional spaces.	23
4.1.	An overall graph of the Random Forest model. [ZZP ⁺ 20]	27
4.2.	Encoder-Decoder architecture	34
4.3.	Encoder-Decoder architecture with canonical attention mechanism	36
4.4.	The essence of Attention mechanism.	37
4.5.	Three-stage calculation Attention process.	38
4.6.	An overall graph of the Encoder-decoder architecture of the Informer model. [ZZP ⁺ 20]	
	41

List of Figures

5.1. The data pipeline overview diagram shows how raw data undergoes a sequence of processing into a training data set and enters the machine learning model.	43
5.2. An example format of the input distance matrix. the basic information is obtained from [Nan].	45
5.3. A demonstration shows how to yield the complete output file.	45
5.4. When the window size is 3, the RMSE of the Gaussian Process changes with PCA = [5, 10, 20, 30, 40] based on the Betti number, Betti sequence, and Filtration dataset.	46
5.5. The Informer model encoder architecture	47
5.6. RMSE of 5 models with window size 3, 5, 7 based predictions of 2017 Bitcoin prices.	48
5.7. RMSE of 5 models with window size 3, 5, 7 based predictions of 2020 Bitcoin prices.	48
5.8. Gaussian Process (GP) based regression performance in 2017.	49
5.9. Gaussian Process (GP) based regression performance in 2020.	49
5.10. Random Forest (RF) based regression performance in 2020.	50
5.11. Gaussian Process (GP) based regression performance in 2020.	50
5.12. Elastic net (ENET) based regression performance in 2020.	50
5.13. XGBT based regression performance in 2020.	50
5.14. RMSE of 5 models with window size 3,5,7 based predictions of 2020 Ethereum prices.	51
5.15. The distribution of node on Bitcoin and Ethereum blockchain dataset, the data of left graph are from Bitcoin Occurrence matrix, the right's data are from Ethereum Occurrence matrix.	52
A.1. The covariance matrix of the transaction volume of the top 8 cryptocurrencies with the highest market capitalization.	59
A.2. Time series of daily log price, total transactions, average Betti-0 and Betti-1 numbers in 2020.	60
A.3. Time series of daily log price, total transactions, average Betti-0 and Betti-1 numbers in 2017.	61
A.4. Random Forest (RF) model performance based on 2017 Bitcoin dataset.	61
A.5. Random Forest (RF) model performance in 2020.	62
A.6. XGBT model performance based on 2017 Bitcoin dataset.	62
A.7. XGBT model performance based on 2020 Bitcoin dataset.	62
A.8. ENET model performance based on 2017 Bitcoin dataset.	62
A.9. ENET model performance based on 2020 Bitcoin dataset.	63
A.10. Informer based baseline performance based on 2017 Bitcoin dataset.	63
A.11. Informer based baseline performance based on 2020 Bitcoin dataset.	63
A.12. % Change (decrease) in RMSE compared to the baseline model.	64
A.13. Price prediction for 2016 with 1 day horizon.	65
A.14. Price prediction for 2016 with 5 day horizon.	65
A.15. Price prediction for 2016 with 10 day horizon.	65
A.16. Price prediction for 2016 with 20 day horizon.	65

Bibliography

- [AAA14] ARIYO, Adebiyi A. ; ADEWUMI, Adewumi O. ; AXO, Charles K.: Stock price prediction using the ARIMA model. In: *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation* IEEE, 2014, S. 106–112
- [AAG⁺19] ABAY, Nazmiye C. ; AKCORA, Cuneyt G. ; GEL, Yulia R. ; ISLAMBEKOV, Umar D. ; KANTARCIOLU, Murat ; TIAN, Yahui ; THURAISINGHAM, Bhavani: Chainnet: Learning on blockchain graphs with topological features. In: *arXiv preprint arXiv:1908.06971* (2019)
- [ADGK18] AKCORA, Cuneyt G. ; DEY, Asim K. ; GEL, Yulia R. ; KANTARCIOLU, Murat: Forecasting bitcoin price with graph chainlets. In: *Pacific-Asia conference on knowledge discovery and data mining* Springer, 2018, S. 765–776
- [B⁺13] BUTERIN, Vitalik u. a.: Ethereum white paper. In: *GitHub repository* 1 (2013), S. 22–23
- [BCB14] BAHDANAU, Dzmitry ; CHO, Kyunghyun ; BENGIO, Yoshua: Neural machine translation by jointly learning to align and translate. In: *arXiv preprint arXiv:1409.0473* (2014)
- [BKC17] BADRINARAYANAN, Vijay ; KENDALL, Alex ; CIPOLLA, Roberto: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. In: *IEEE transactions on pattern analysis and machine intelligence* 39 (2017), Nr. 12, S. 2481–2495
- [Blo] BLOCKCHAIN.COM: *Blockchain Data API*. <https://www.blockchain.com/api/blockchainapi>
- [BPC20] BELTAGY, IZ ; PETERS, Matthew E. ; COHAN, Arman: Longformer: The long-document transformer. In: *arXiv preprint arXiv:2004.05150* (2020)
- [Bre01] BREIMAN, Leo: Random forests. In: *Machine learning* 45 (2001), Nr. 1, S. 5–32
- [Car09] CARLSSON, Gunnar: Topology and data. In: *Bulletin of the American Mathematical Society* 46 (2009), Nr. 2, S. 255–308
- [CG16] CHEN, Tianqi ; GUESTRIN, Carlos: Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, S. 785–794
- [CGRS19] CHILD, Rewon ; GRAY, Scott ; RADFORD, Alec ; SUTSKEVER, Ilya: Generating long sequences with sparse transformers. In: *arXiv preprint arXiv:1904.10509* (2019)
- [coia] COINCODEX: *Live Crypto Prices and Cryptocurrency Market Cap*. <https://coincodex.com/>

Bibliography

- [Coib] COINMARKETCAP: *Top Cryptocurrency Exchanges Ranked By Volume*. <https://coinmarketcap.com/rankings/exchanges/>
- [com] COMPANIESMARKETCAP: *The Global Largest Companies by Market Cap*. <https://companiesmarketcap.com/>
- [CVMBB14] CHO, Kyunghyun ; VAN MERRIËNBOER, Bart ; BAHDANAU, Dzmitry ; BENGIO, Yoshua: On the properties of neural machine translation: Encoder-decoder approaches. In: *arXiv preprint arXiv:1409.1259* (2014)
- [Dey16] DEY, Ayon: Machine learning algorithms: a review. In: *International Journal of Computer Science and Information Technologies* 7 (2016), Nr. 3, S. 1174–1179
- [FM99] FREUND, Yoav ; MASON, Llew: The alternating decision tree learning algorithm. In: *icml* Bd. 99 Citeseer, 1999, S. 124–133
- [Fre77] FREEMAN, Linton C.: A set of measures of centrality based on betweenness. In: *Sociometry* (1977), S. 35–41
- [GA15] GREAVES, Alex ; AU, Benjamin: Using the bitcoin transaction graph to predict the price of bitcoin. In: *No Data* (2015)
- [GEN21] GENERAL, NEW YORK STATE A.: Attorney General James Ends Virtual Currency Trading Platform Bitfinex’s Illegal Activities in New York. (February 23rd 2021)
- [Gov] GOVERNMENT, United S.: *The Coronavirus Aid, Relief, and Economic Security Act*. <https://home.treasury.gov/policy-issues/coronavirus>
- [GR17] GUPTA, Ankit ; RUSH, Alexander M.: Dilated convolutions for modeling long-distance genomic dependencies. In: *arXiv preprint arXiv:1710.01278* (2017)
- [HF96] HYNDMAN, Rob J. ; FAN, Yanan: Sample quantiles in statistical packages. In: *The American Statistician* 50 (1996), Nr. 4, S. 361–365
- [Inf] INFURA: *The World’s Most Powerful Blockchain Development Suite*. <https://infura.io/>
- [Jol05] JOLLIFFE, Ian: Principal component analysis. In: *Encyclopedia of statistics in behavioral science* (2005)
- [Lim] LIMITED., Fusion M.: *investing.com*. <https://www.investing.com/>
- [LJX⁺19] LI, Shiyang ; JIN, Xiaoyong ; XUAN, Yao ; ZHOU, Xiyou ; CHEN, Wenhui ; WANG, Yu-Xiang ; YAN, Xifeng: Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In: *arXiv preprint arXiv:1907.00235* (2019)
- [MB19] MOTAMED, Amir P. ; BAHRAK, Behnam: Quantitative analysis of cryptocurrencies transaction graph. In: *Applied Network Science* 4 (2019), Nr. 1, S. 1–21
- [MN13] MISCHAIKOW, Konstantin ; NANDA, Vidit: Morse theory for filtrations and efficient computation of persistent homology. In: *Discrete & Computational Geometry* 50 (2013), Nr. 2, S. 330–353

- [Nak19] NAKAMOTO, Satoshi: Bitcoin: A peer-to-peer electronic cash system / Manubot. 2019.
– Forschungsbericht
- [Nan] NANDA, Vudit: *Perseus, the Persistent Homology Software*. <http://people.maths.ox.ac.uk/nanda/perseus/>
- [New05] NEWMAN, Mark E.: A measure of betweenness centrality based on random walks. In: *Social networks* 27 (2005), Nr. 1, S. 39–54
- [PIL06] PRASAD, Anantha M. ; IVERSON, Louis R. ; LIAW, Andy: Newer classification and regression tree techniques: bagging and random forests for ecological prediction. In: *Ecosystems* 9 (2006), Nr. 2, S. 181–199
- [Ras03] RASMUSSEN, Carl E.: Gaussian processes in machine learning. In: *Summer school on machine learning* Springer, 2003, S. 63–71
- [Reu05] REUTER, Peter: *Chasing dirty money: The fight against money laundering*. Peterson Institute, 2005
- [RS13] RON, Dorit ; SHAMIR, Adi: Quantitative analysis of the full bitcoin transaction graph. In: *International Conference on Financial Cryptography and Data Security* Springer, 2013, S. 6–24
- [Sch19] SCHETSEN, Anouk van: Impact of graph-based features on Bitcoin prices. (2019)
- [Swa14] SWANSON, Tim: *Learning from Bitcoin’s past to improve its future*. 2014
- [Tap] TAPPE, Anneken: *Bitcoin? Ethereum? Dogecoin? Your guide to the biggest names in crypto*. <https://edition.cnn.com/2021/04/22/investing/cryptocurrency-guide-top-five-bitcoin-ethereum/index.html>
- [UNO11] UNODC: UNODC estimates that criminals may have laundered US \$1.6 trillion in 2009. (2011)
- [VSP⁺17] VASWANI, Ashish ; SHAZER, Noam ; PARMAR, Niki ; USZKOREIT, Jakob ; JONES, Llion ; GOMEZ, Aidan N. ; KAISER, Lukasz ; POLOSUKHIN, Illia: Attention is all you need. In: *arXiv preprint arXiv:1706.03762* (2017)
- [WR96] WILLIAMS, Christopher K. ; RASMUSSEN, Carl E.: Gaussian processes for regression. (1996)
- [YKF17] YU, Fisher ; KOLTUN, Vladlen ; FUNKHOUSER, Thomas: Dilated residual networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, S. 472–480
- [ZH05] ZOU, Hui ; HASTIE, Trevor: Regularization and variable selection via the elastic net. In: *Journal of the royal statistical society: series B (statistical methodology)* 67 (2005), Nr. 2, S. 301–320
- [Zom10] ZOMORODIAN, Afra: Fast construction of the Vietoris-Rips complex. In: *Computers & Graphics* 34 (2010), Nr. 3, S. 263–271

Bibliography

- [ZZP⁺20] ZHOU, Haoyi ; ZHANG, Shanghang ; PENG, Jieqi ; ZHANG, Shuai ; LI, Jianxin ; XIONG, Hui ; ZHANG, Wancai: Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In: *arXiv preprint arXiv:2012.07436* (2020)
- [ZZWD20] ZHENG, Peilin ; ZHENG, Zibin ; WU, Jiajing ; DAI, Hong-Ning: Xblock-ETH: extracting and exploring blockchain data from Ethereum. In: *IEEE Open Journal of the Computer Society* 1 (2020), S. 95–106