LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

FAKULTÄT FÜR MATHEMATIK, INFORMATIK UND STATISTIK
INSTITUT FÜR INFORMATIK

LEHRSTUHL FÜR DATENBANKSYSTEME
UND DATA MINING

On the topological property of dynamic transaction graph

Master Thesis

Yuhao Wang

13.08.2021

DBS

Scientific Supervisor: **Dr. Yunpu Ma**
Responsible professor: **Prof. Dr. Volker Tresp**
Company**: secunet Security Networks AG**

Contents

Introduction

Motivation

- Research the laws of market transaction data through topological data analysis in high dimensional space.
- Explore the changing trends of the specific cryptocurrency transaction and the overall cryptocurrency market.
- Provide data support for market investment and help establish risk assessment models.

Introduction

Problem Statement

1. Examine the dynamic property of topological features on different blockchains.
2. Research the persistent effects of the topological properties on the Blockchain after the explosion of cryptocurrency trading.
3. Explore the benefit of the Informer on the dynamic transaction graph.
4. Examine the interaction between various cryptocurrency blockchains..

Introduction

Blockchain

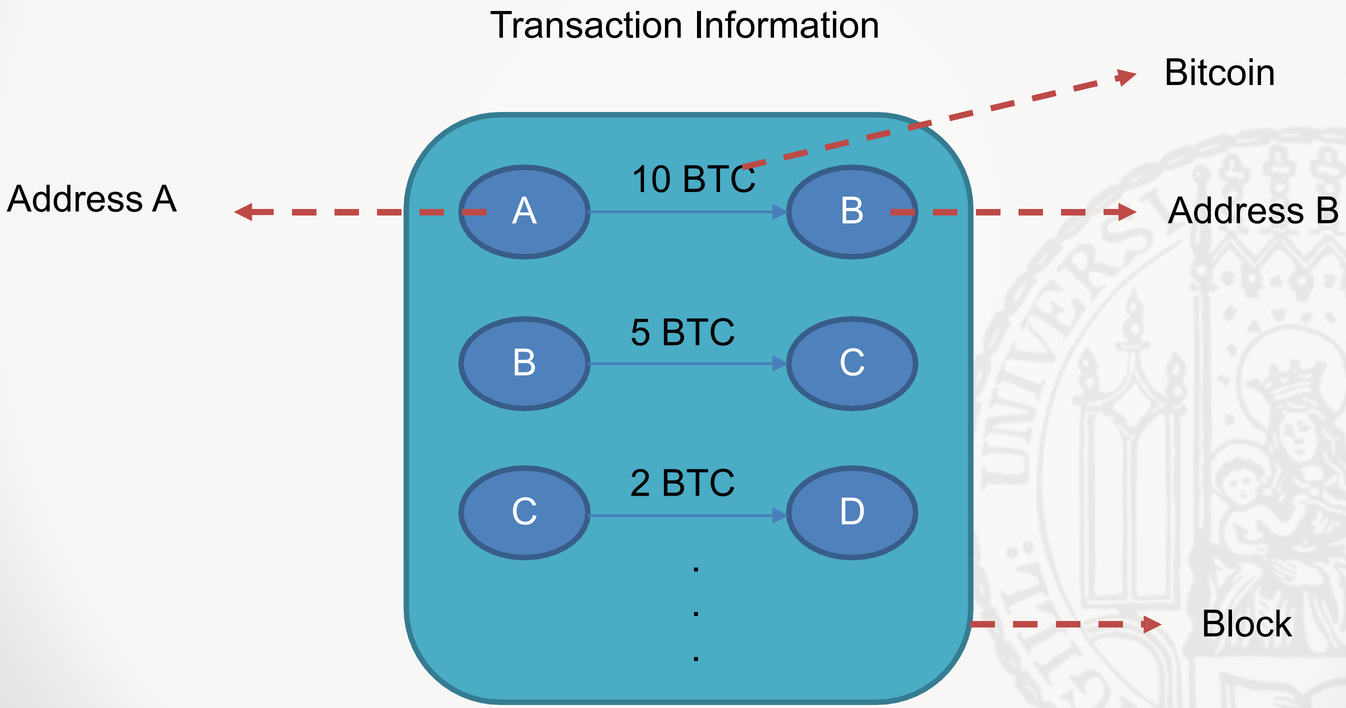


Fig. 1: A example of Block

Introduction

Blockchain

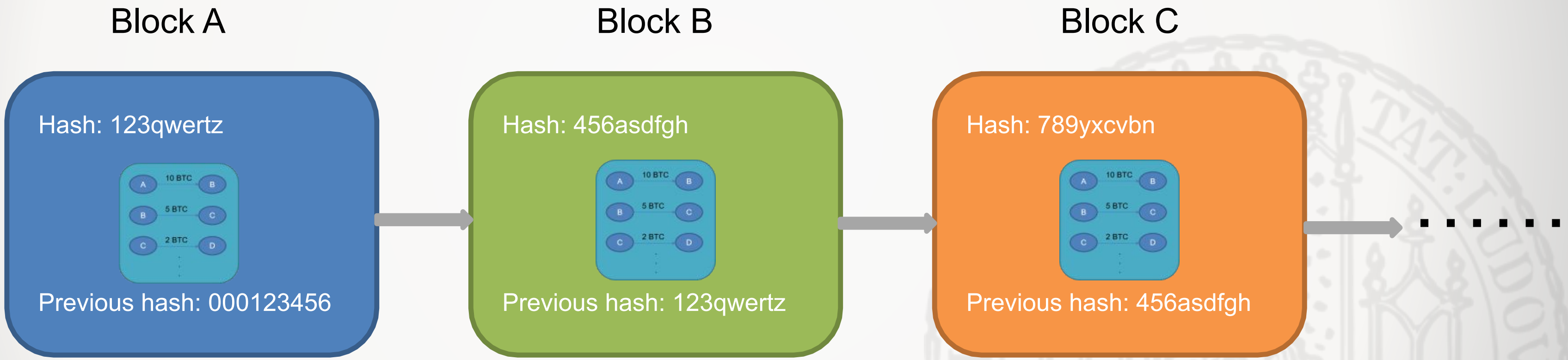


Fig. 2: A example of a BlockChain with three blocks; A,B and C. The system produces a new block every 10 minutes.

Methodology

A. Learning Graph Representations (Bitcoin)

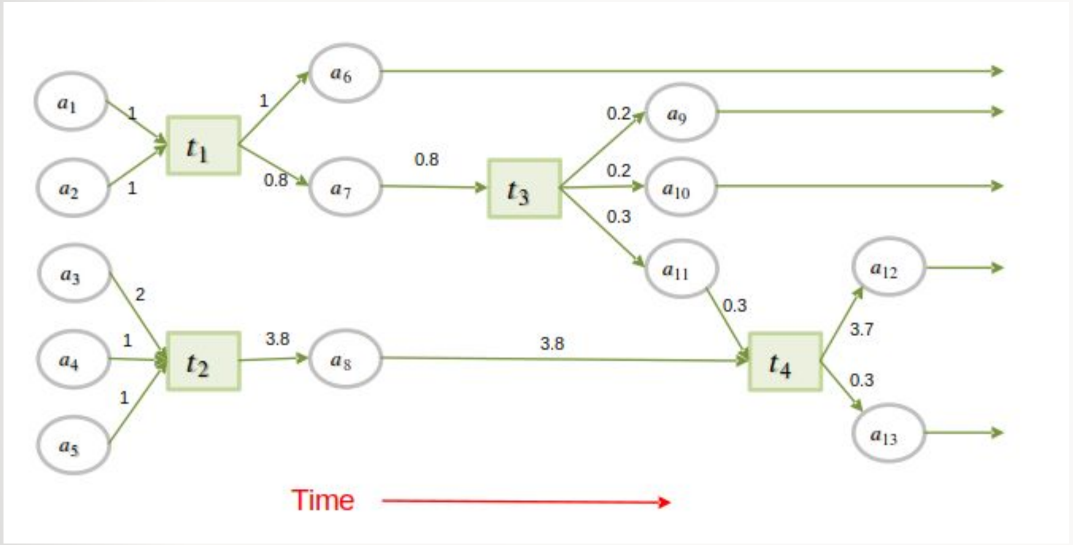


Fig. 3: A transaction-address graph representation of the Bitcoin network.

Notation:

- a: address
- t: transaction
- the value between a and t: the number of bitcoins transferred

1. Cuneyt el all., Forecasting bitcoin price with graph chainlets, PAKDD, 2018

Methodology

A. Learning Graph Representations

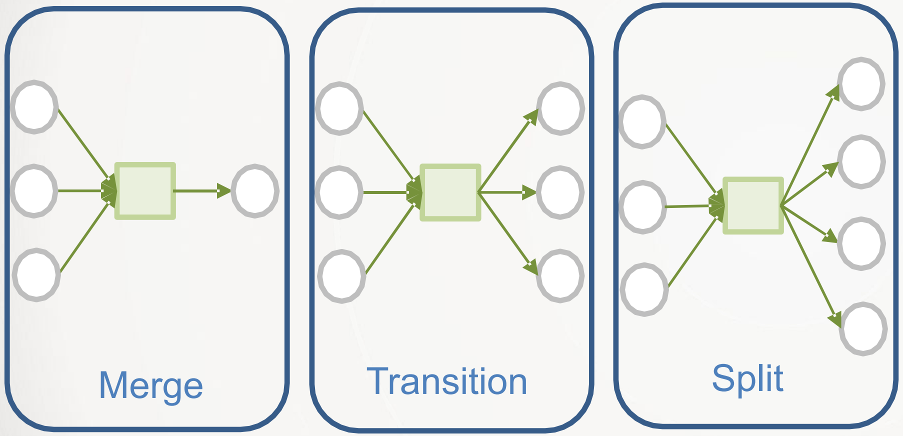


Fig. 4: Transaction patterns of the above figure can be divided into the following three types: Merge ($C_{3\to1}$), Transition ($C_{3\to3}$), Split ($C_{3\to4}$)

Notation:

$C_{x\to y}$ (chainlet [1]) refers to x inputs and y outputs:

- Merge: x > y
- Transaction: x = y
- Split: x < y

1. Cuneyt el all., Forecasting bitcoin price with graph chainlets, PAKDD, 2018

Methodology

A. Learning Graph Representations

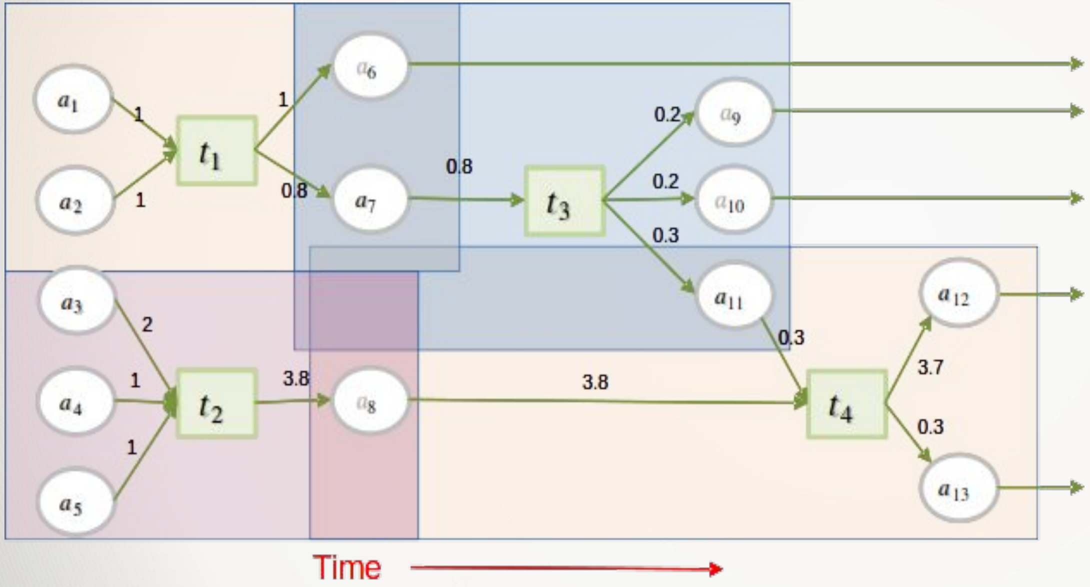


Fig. 5: A transaction-address graph representation of the Bitcoin network. a refers to address, t is transaction, the value between a and t is the number of bitcoins transferred.

1. Occurrence and Amount Matrices:

$$O = \begin{bmatrix} 0 & 0 & \boxed{1} \\ 0 & \boxed{2} & 0 \\ \boxed{1} & 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & \boxed{0.8} \\ 0 & \boxed{6.1} & 0 \\ \boxed{4} & 0 & 0 \end{bmatrix}$$

Occurrence matrix:

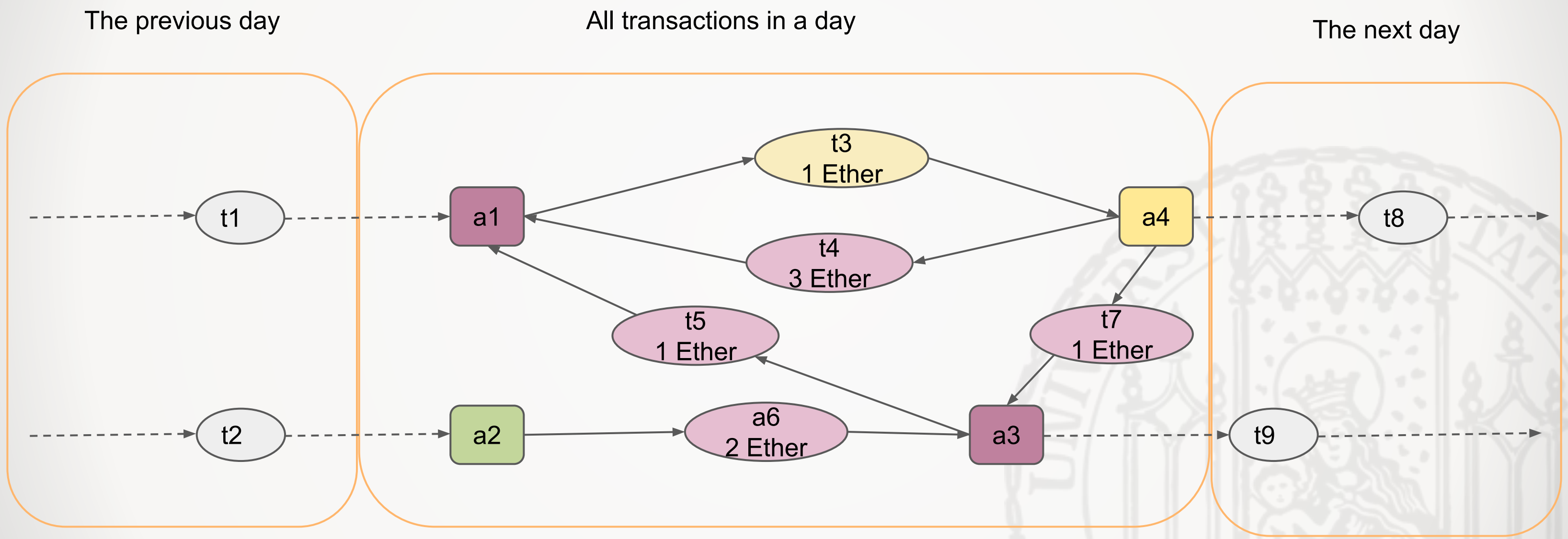
- Row: the amount of the input
- Column: the amount of the output
- Value: the amount of the transaction with specific input and output

Amount matrix:

- Row: the amount of the input
- Column: the amount of the output
- Value: the value of the transferred BTC with specific input and output

Methodology

A. Learning Graph Representations (ETH)



O: $\begin{bmatrix} 0 & 0 & \boxed{2} \\ \boxed{1} & 0 & 0 \\ 0 & \boxed{1} & 0 \end{bmatrix}$

A: $\begin{bmatrix} 0 & 0 & \boxed{7} \\ 0 & 0 & 0 \\ 0 & \boxed{1} & 0 \end{bmatrix}$

Notation:

- a: address
- t: transaction
- the value between a and t: the amount of the Ether

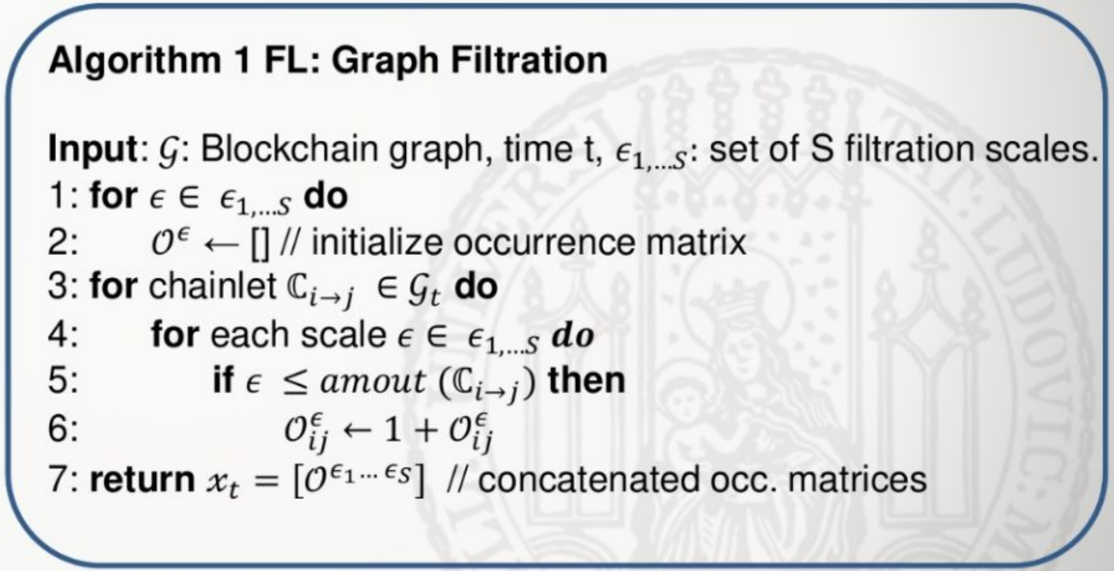
Fig. 6: A transaction-address graph representation of the Ethereum network.

Methodology

A. Learning Graph Representations

Graph Filtration (FL):

- Given the amount and occurrence information, a natural combination of them entails filtering the occurrence matrix with user defined thresholds on amounts, or filtering the amount matrix with user defined thresholds on occurrences.



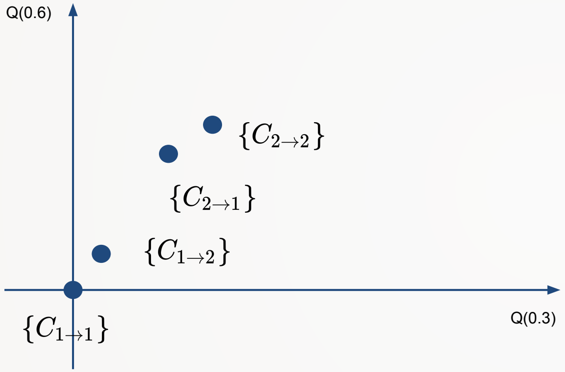
Algorithm 1 FL: Graph Filtration

```
Input: $\mathcal{G}$: Blockchain graph, time t, $\epsilon_{1,\ldots S}$: set of S filtration scales.
1: for $\epsilon \in \epsilon_{1,\ldots S}$ do
2:     $\mathcal{O}^{\epsilon} \leftarrow []$ // initialize occurrence matrix
3: for chainlet $\mathbb{C}_{i\to j} \in \mathcal{G}_t$ do
4:     for each scale $\epsilon \in \epsilon_{1,\ldots S}$ do
5:         if $\epsilon \leq amout\ (\mathbb{C}_{i\to j})$ then
6:             $\mathcal{O}^{\epsilon}_{ij} \leftarrow 1 + \mathcal{O}^{\epsilon}_{ij}$
7: return $x_t = [\mathcal{O}^{\epsilon_1 \cdots \epsilon_S}]$ // concatenated occ. matrices
```

Methodology

B. Learning Topological Representations

$$\left\{\begin{array}{cc} \textit{chainlet} & \textit{amount array} \\ C_{1\to1} & [0] \\ C_{1\to2} & [0.8] \\ C_{2\to1} & [2, 4.1] \\ C_{2\to2} & [4] \end{array}\right\} \Rightarrow$$



$\Rightarrow$

		$C_{1\to1}$	$C_{1\to2}$	$C_{2\to1}$	$C_{2\to2}$
		$(0,0)$	$(0.8,0.8)$	$(2.63,3.26)$	$(4,4)$
$C_{1\to1}$	$(0,0)$	0	1.28	17.54	32
$C_{1\to2}$	$(0.8,0.8)$	1.28	0	9.4	20.48
$C_{2\to1}$	$(2.63,3.26)$	17.54	9.4	0	2.42
$C_{2\to2}$	$(4,4)$	32	20.48	2.42	0

1. A k-th quantile, k = 0,1,2,…,q is the amount Q(k), when we set the quantile [0.3,0.6] of $\{C_{2\to2}\}$ is (2.63, 3.26), for $\{C_{3\to1}\}$ is (4, 4), for $\{C_{1\to3}\}$ is (0.8, 0.8), for $\{C_{1\to1}, C_{1\to2}, C_{2\to1}, C_{2\to3}, C_{3\to2}, C_{3\to3}\}$ is (0, 0), then we can use the middle two-dimensional coordinate system to display all chainlets.
2. Calculate the distance of a pair of chainlets in 2-dimension space to build a distance matrix (right figure):

$$d_{ij} = \sqrt{\sum_{k=0}^{q}[Q_i(k) - Q_j(k)]^2} \Rightarrow d_{c_{2\to2},c_{3\to1}} = \sqrt{(2.63-4)^2 + (3.26-4)^2} = 2.42$$

Methodology

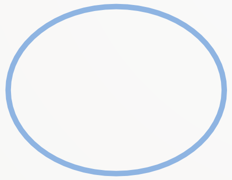
B. Learning Topological Representations

Betti number:

the *k*th Betti number refers to the number of *k*-dimensional *holes* on a topological surface.

Examples:

Betti-0 (b_0) is the number of connected components;
Betti-1 (b_1) is the number of one-dimensional or "circular" holes;
Betti-2 (b_2) is the number of two-dimensional "voids" or "cavities".



$b_0 = 1$ 0D loop	$b_0 = 1$ 0D loop	$b_0 = 1$ 0D loop	$b_0 = 1$ 0D loop
$b_1 = 0$ 1D loop	$b_1 = 1$ 1D loop	$b_1 = 2$ 1D loop	$b_1 = 0$ 1D loop
$b_2 = 0$ 2D loop	$b_2 = 1$ 2D loop	$b_2 = 1$ 2D loop	$b_2 = 1$ 2D loop

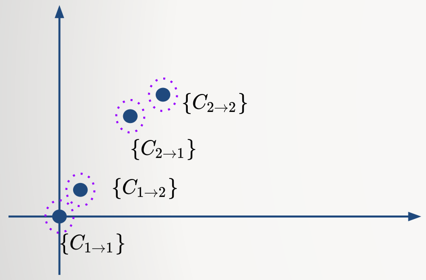
1. Nazmiye el all., ChainNet: Learning on Blockchain Graphs with Topological Features, ICDM, 2019

Methodology

B. Learning Topological Representations

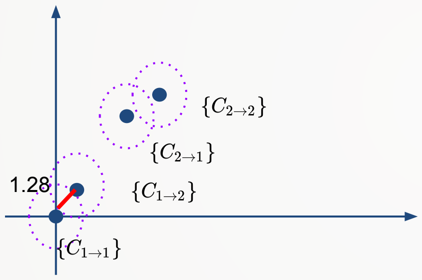
Vietoris-Rips complex (VR): persistent homology

$\epsilon_1 = 1$



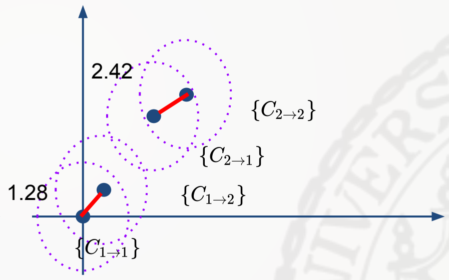
connected components: $\beta_0 = 4$;
"circular" holes: $\beta_1 = 0$

$\epsilon_2 = 2$



connected components: $\beta_0 = 3$;
"circular" holes: $\beta_1 = 0$

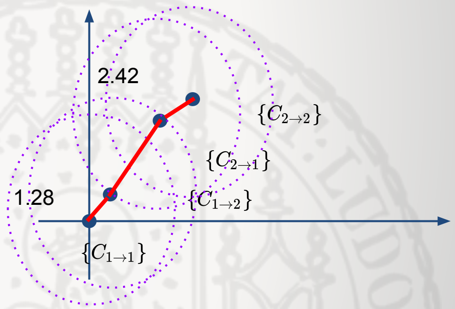
$\epsilon_2 = 2$



connected components: $\beta_0 = 2$;
"circular" holes: $\beta_1 = 0$

…

$\epsilon_{10} = 10$



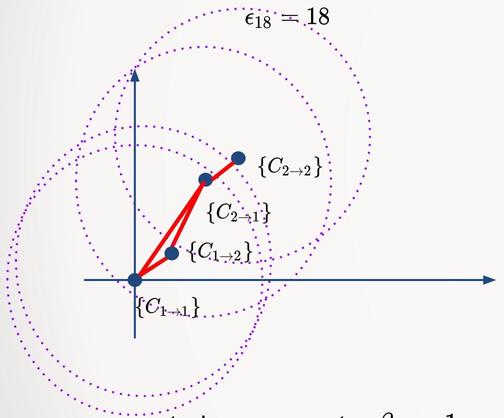
connected components: $\beta_0 = 1$;
"circular" holes: $\beta_1 = 0$

3. For each scale ϵ_k, we build the corresponding VR complex whose 0-simplices are single chainlets and 1-simplices are pairs of chainlets with distance ≤ ϵ_k .

Methodology

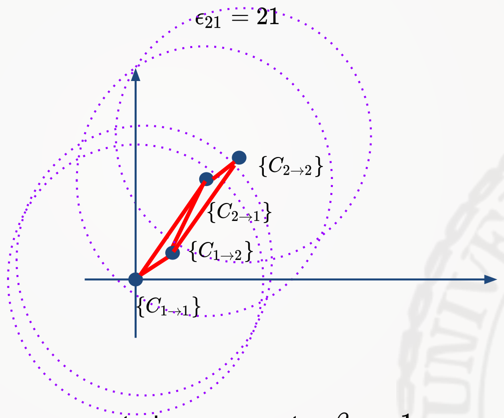
B. Learning Topological Representations

Vietoris-Rips complex: persistent homology



connected components: $\beta_0 = 1$;
"circular" holes: $\beta_1 = 1$

...



connected components: $\beta_0 = 1$;
"circular" holes: $\beta_1 = 2$

...

3. For each scale ϵ_k, we build the corresponding VR complex whose 0-simplices are single chainlets and 1-simplices are pairs of chainlets with distance ≤ ϵ_k .

Methodology

B. Learning Topological Representations

4. Construct a sequence of scale $\epsilon_1 < \epsilon_2 < \ldots < \epsilon_S$ covering a range of distance during the entire 365-day period, The filtration of VR complexes $VR_1 \subseteq VR_2 \ldots \subseteq VR_S$, obtain betti sequence: $x_t = \{\beta_0(\epsilon_1), \ldots, \beta_0(\epsilon_S); \beta_1(\epsilon_1), \ldots, \beta_1(\epsilon_S)\}$

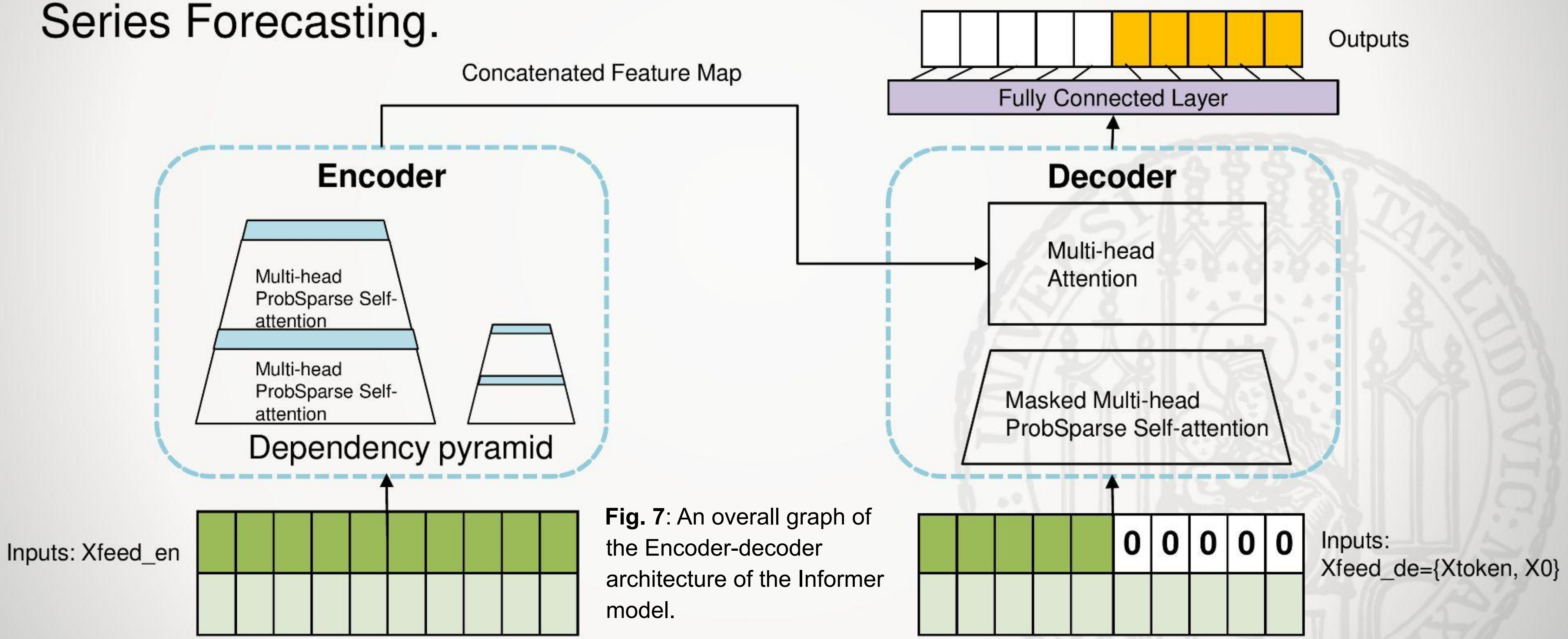
   In the example, $x_t = \{4, 3, 2, \ldots, 1; 0, 0, 0, \ldots, 3\}$

5. Betti derivatives:

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

Methodology

C. Informer: a beyond efficient Transformer for Long Sequence Time-Series Forecasting.



Outputs
Fully Connected Layer
Concatenated Feature Map
Encoder
Multi-head ProbSparse Self-attention
Multi-head ProbSparse Self-attention
Dependency pyramid
Decoder
Multi-head Attention
Masked Multi-head ProbSparse Self-attention
Inputs: Xfeed_en
0 0 0 0 0
Inputs: Xfeed_de={Xtoken, X0}

Fig. 7: An overall graph of the Encoder-decoder architecture of the Informer model.

1. Haoyi el.all, Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. AAAI, 2021.

Methodology

C. Informer

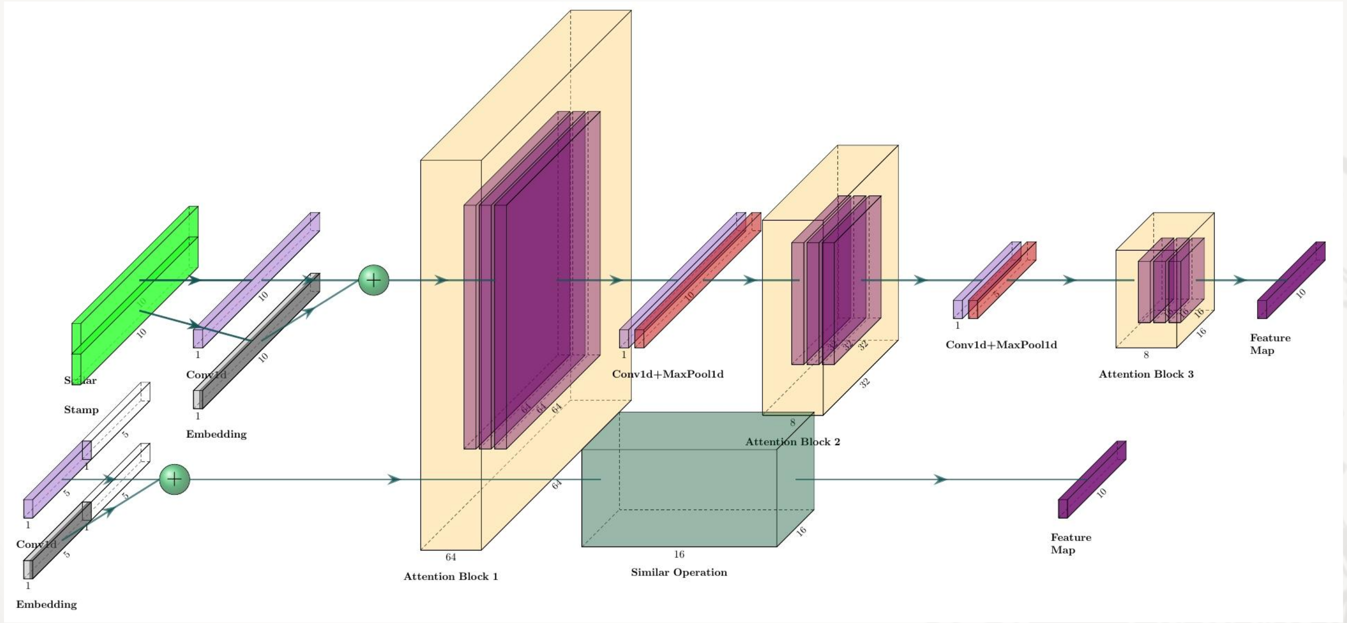


Fig. 8: The structure of Informer model illustrates the cooperation of the 4 convolutional layers, 2 MaxPool, and 3 Multi-heads attention in the Informer's encoder. Each horizontal stack represents an individual encoder replica of the Informer, the upper stack takes the complete input sequence as the primary stack, the second stack uses half slices of the original data, then the ProbSparse 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.

Methodology

C. Transformer vs Informer

Self attention:

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

ProbSparse Self-attention:

$$A(Q,K,V) = Softmax(\frac{\overline{Q}K^T}{\sqrt{d}})V$$

$\overline{Q}$ 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 and $u = c \cdot lnL_Q$, then the algorithm complexity of the dot-product is $O(lnL_Q)$ and the memory usage maintains $O(L_K lnL_Q)$.

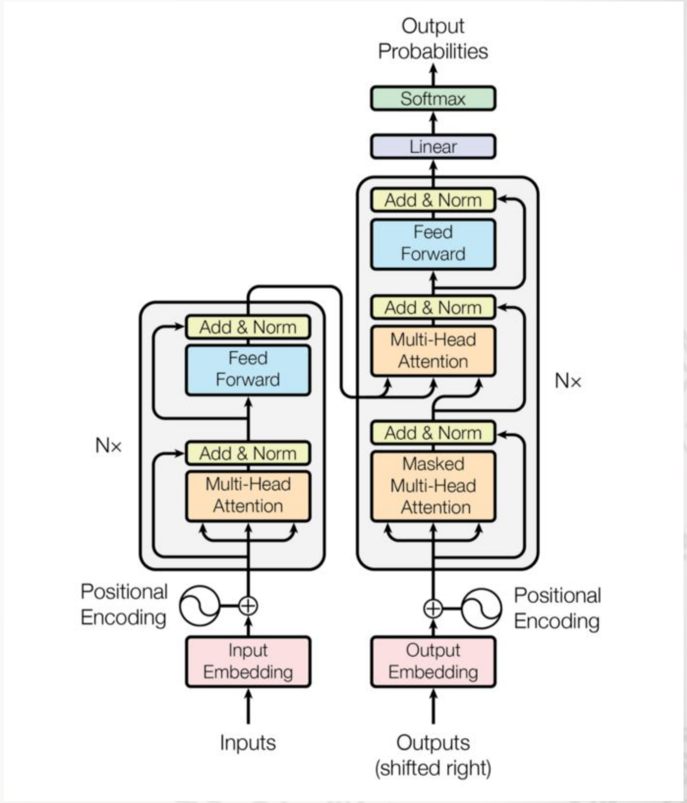


Fig 9: The Transformer - model architecture.

Experiment

Dataset

1. Bitcoin blockchain transaction data

   Source: Blockchain Data API(https://www.blockchain.com/api/blockchain_api)

2. Ethereum blockchain transaction data

   Source: Infura IPFS API(https://infura.io/docs/ipfs)

3. The volume, market capitalization, price of the primary cryptocurrency.

   Source: web scrape(https://coinmarketcap.com/)

Experiment

Feature Engineering

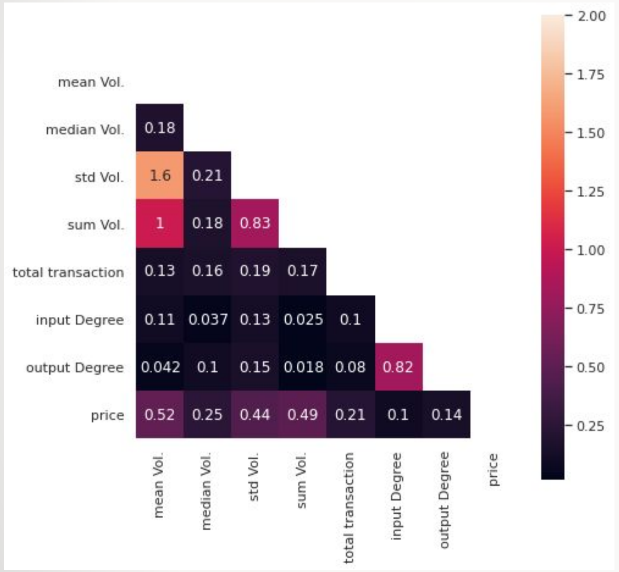


Fig. 10:The covariance matrix of Bitcoin blockchain various properties.

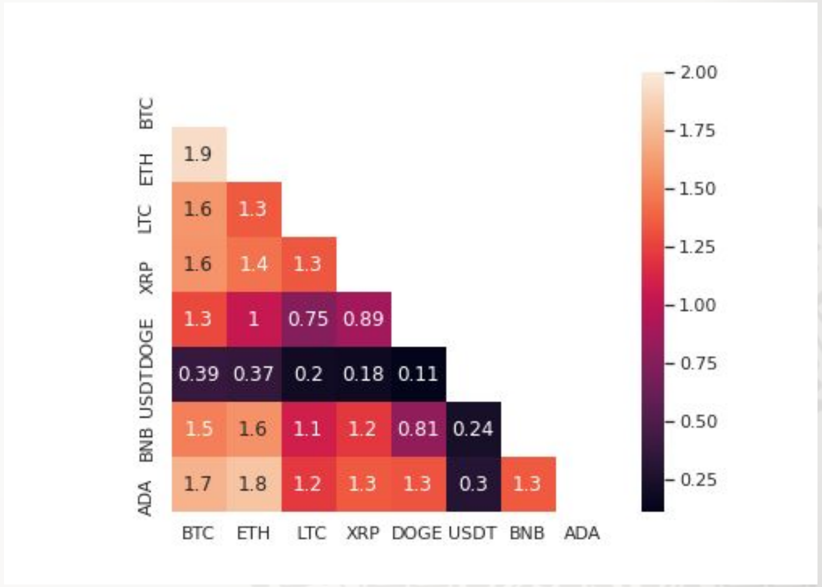
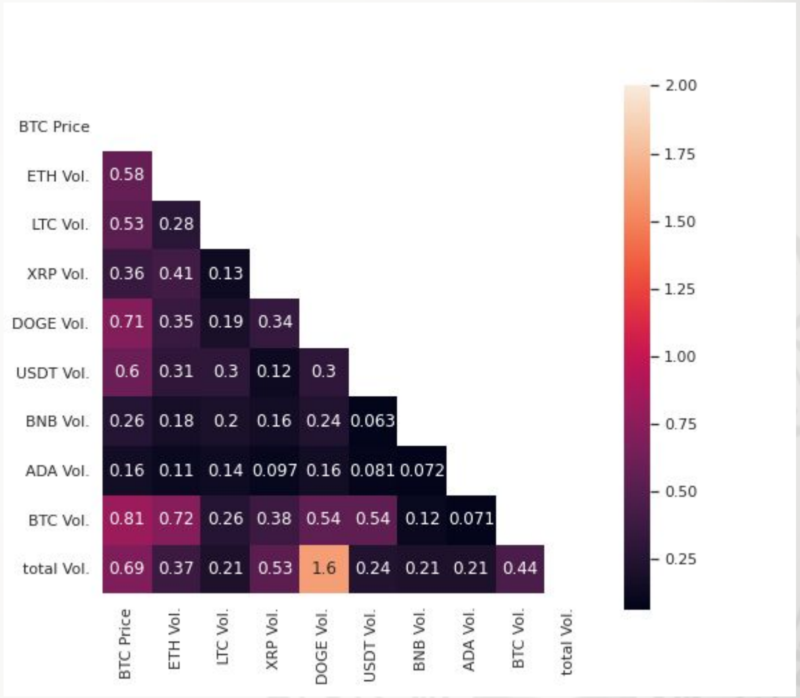


Fig. 11: The covariance matrix of the top 8 cryptocurrencies price by market capitalization.

Experiment

Feature Engineering

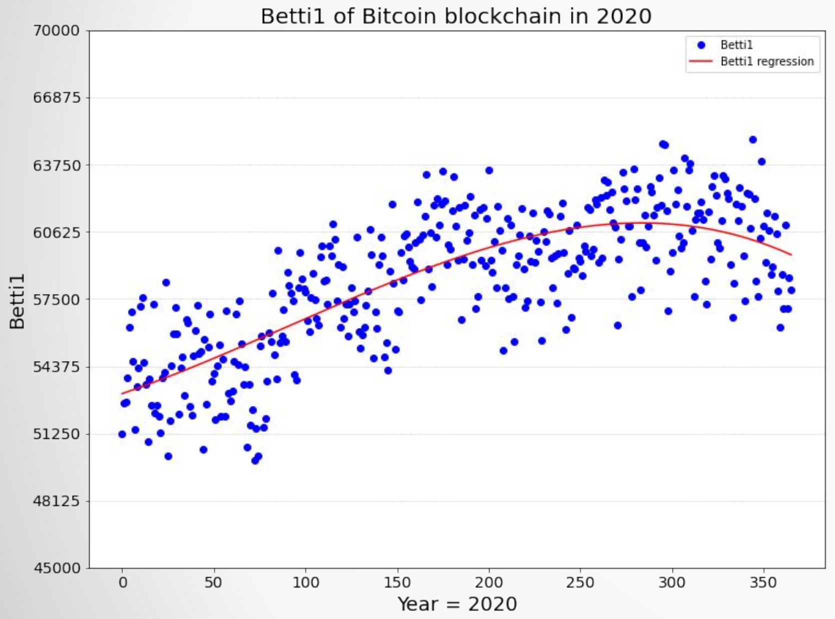
Fig. 12: The covariance matrix of the transaction volume of the top 8 cryptocurrencies with the highest market capitalization.



	BTC Price	ETH Vol.	LTC Vol.	XRP Vol.	DOGE Vol.	USDT Vol.	BNB Vol.	ADA Vol.	BTC Vol.	total Vol.
BTC Price										
ETH Vol.	0.58									
LTC Vol.	0.53	0.28								
XRP Vol.	0.36	0.41	0.13							
DOGE Vol.	0.71	0.35	0.19	0.34						
USDT Vol.	0.6	0.31	0.3	0.12	0.3					
BNB Vol.	0.26	0.18	0.2	0.16	0.24	0.063				
ADA Vol.	0.16	0.11	0.14	0.097	0.16	0.081	0.072			
BTC Vol.	0.81	0.72	0.26	0.38	0.54	0.54	0.12	0.071		
total Vol.	0.69	0.37	0.21	0.53	1.6	0.24	0.21	0.21	0.44	

Experiment

Feature Engineering



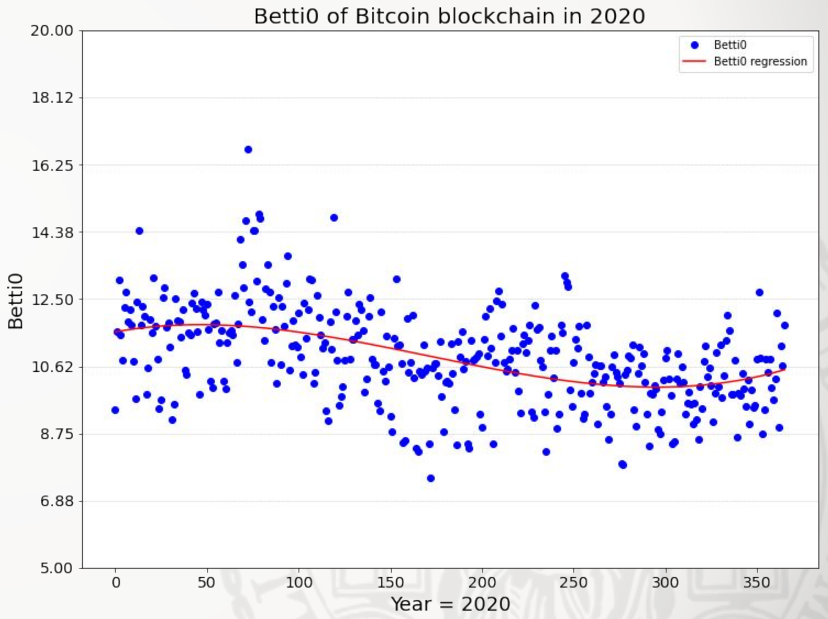
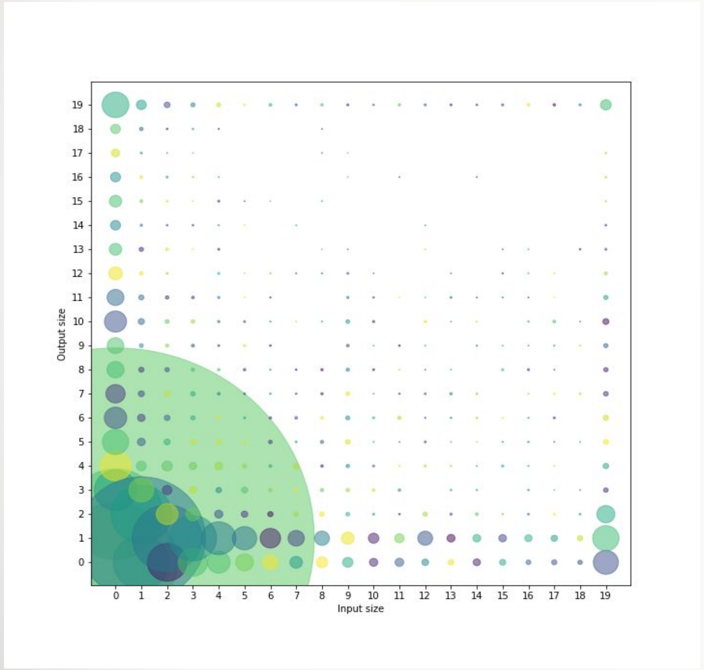


Fig. 13: The picture on the left is the 2020 Bitcoin blockchain Betti-0, and the picture on the right is the 2020 Bitcoin blockchain Betti-1.

Experiment

Bitcoin vs Ethereum



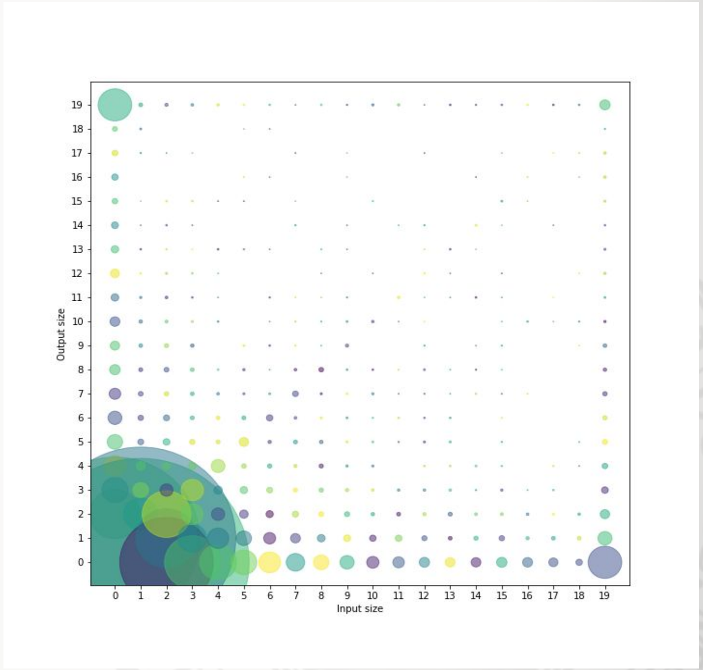


Fig. 14: The distribution of node on Bitcoin and Ethereum blockchain dataset, the data of left graph are from Bitcoin Occurrence matrix, the right's data are from Ethereum Occurrence matrix.

Experiment

Sliding Window

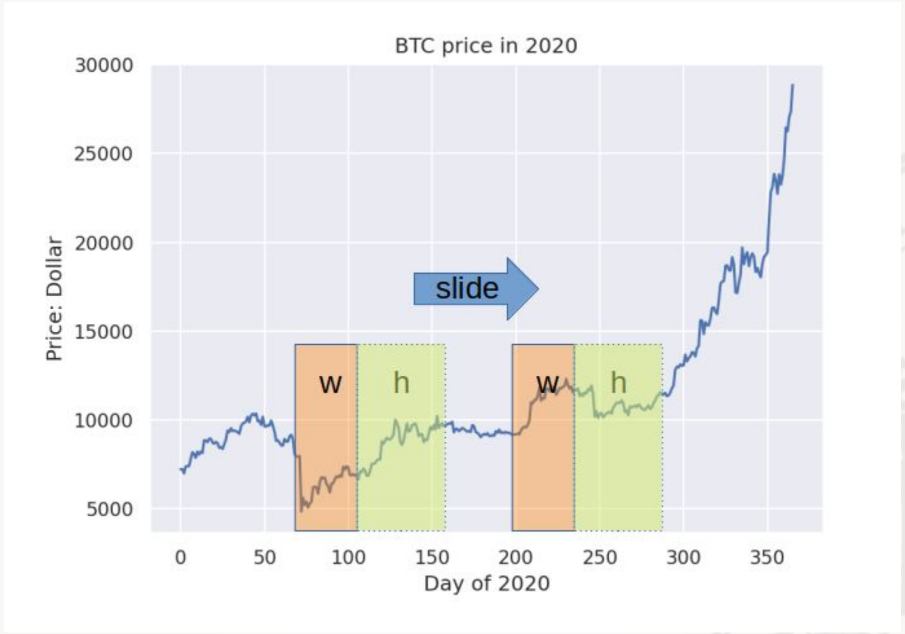


Fig. 15: Visualization of a sliding window, with window size w and h the prediction horizons.

Experiment

Data Pipeline

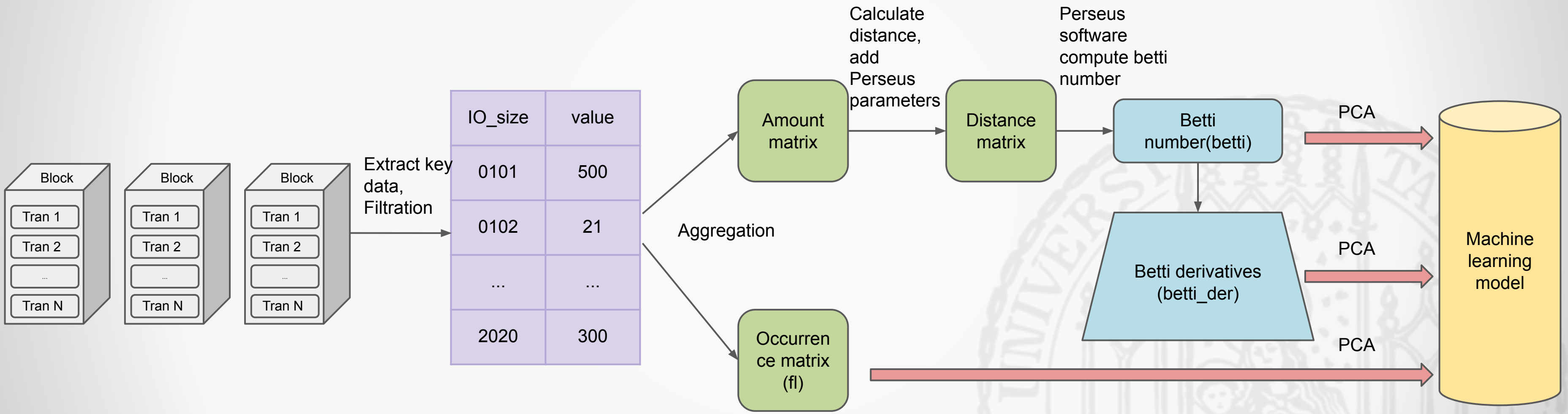


Fig. 16: The data pipeline overview diagram shows how raw data undergoes a sequence of processing into a training data set and enters the machine learning model.

Experiment

Distance matrix

4			
1	1	100	1
0	1.28	17.54	32
1.28	0	9.4	2.42
17.54	9.4	0	2.42
32	20.48	2.42	0

4 : this is the number of rows/columns in the symmetric distance matrix.

1, 1, 100, 1 : initial threshold distance g = 1, step size s = 1, number of steps N = 100 and dimension cap C = 1.

0 1.28 17.54 32 : distance from $C_{1\to 1}$ to other chainlets.

...

Fig. 17: An example format of the input distance matrix. the basic information is obtained from Perseus software.

Experiment

Betti number

Perseus software output file

0	4	0
1	3	0
2	2	0
9	1	0
17	1	1
20	1	2
31	1	3

Fill in the missing content.

Complete file

	β_0	β_1
0	4	0
1	3	0
2	2	0
3	2	0
4	2	0
5	2	0
...	...	...
17	1	1
18	1	1
...	...	...
31	1	3

Fig. 18: A demonstration shows how to yield the complete output file.

Evaluation

- 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, y_t and $\hat{y}_t$ represent the actual price and predict price separated at the day t.

- 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.

Evaluation

Gaussian Process(Bitcoin)

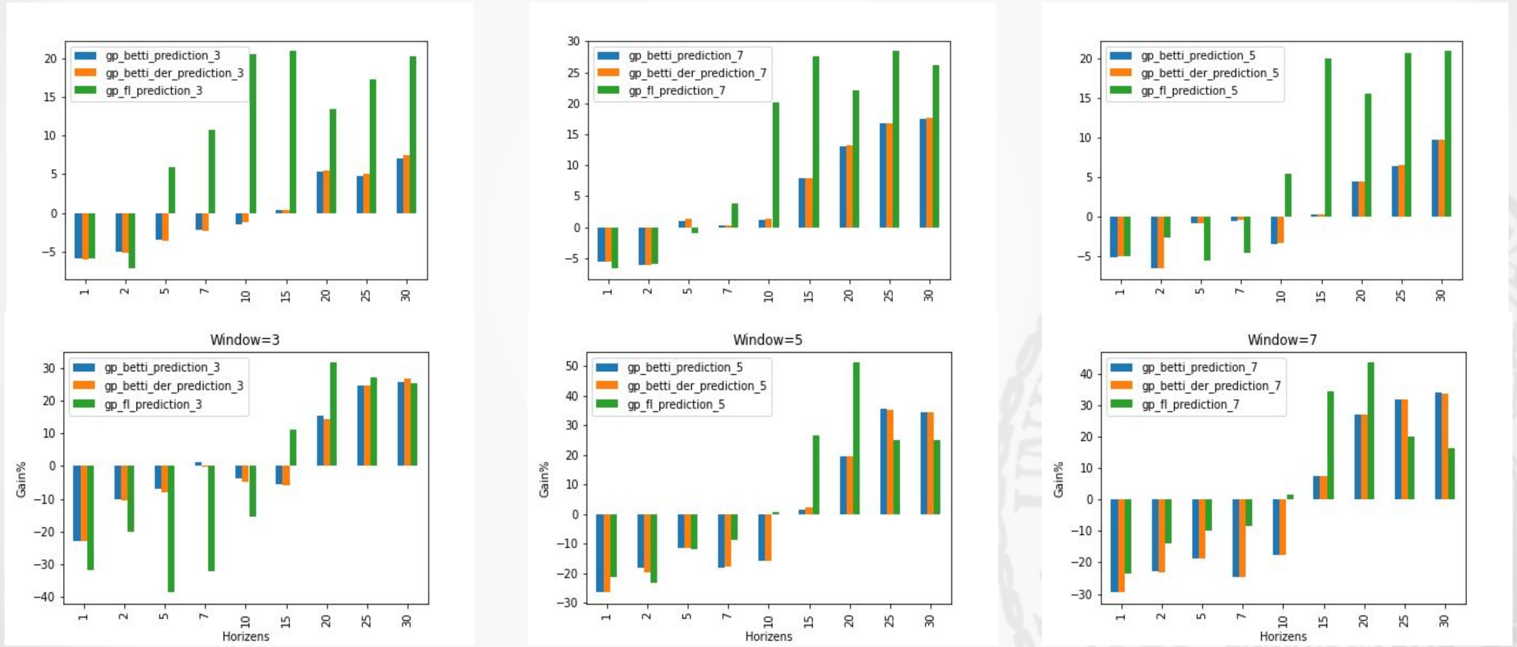


Fig. 19: The figure in the first row is generated by the 2017 Bitcoin transaction graph data, and the figure in the second row is generated by the Bitcoin transaction graph data in 2020. gp_betti_prediction_3 represents the Gaussian process model, the Betti number data set, and the window size is 3. betti_der is the Betti derivative data set, and fl is the filtration data set.

Evaluation

XGBT(Bitcoin)

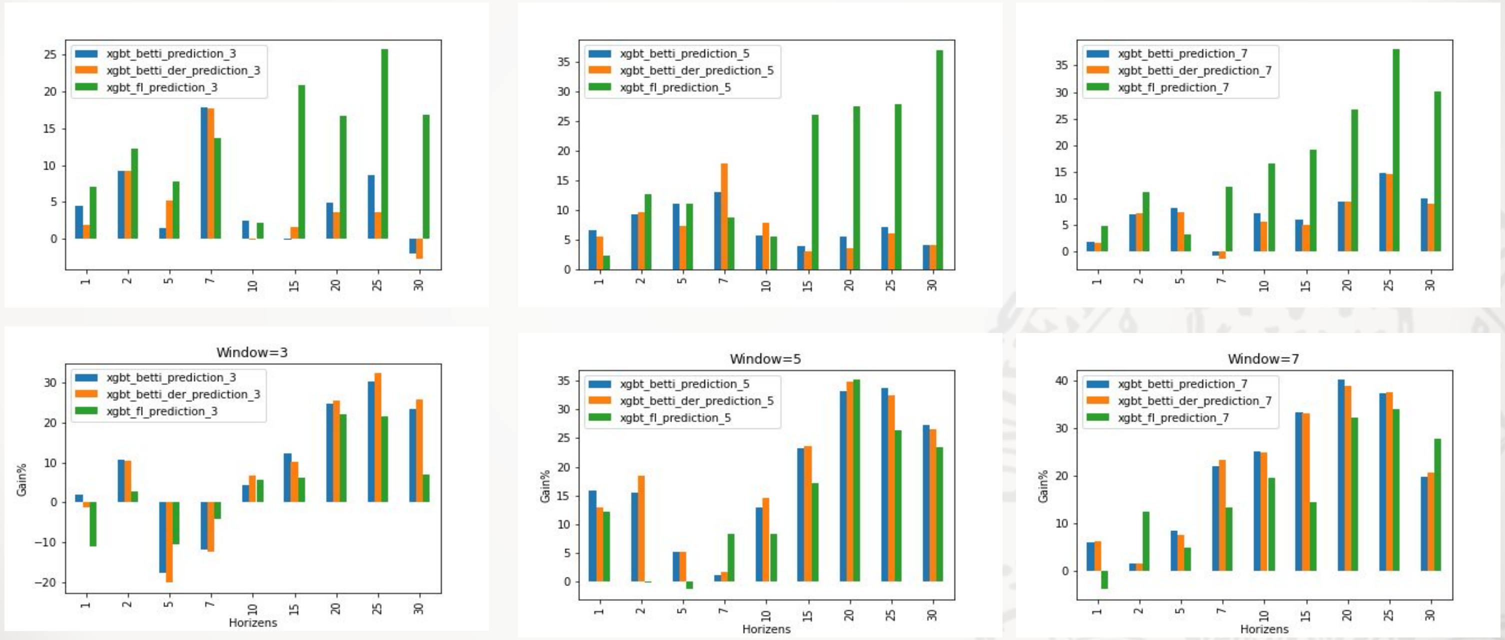
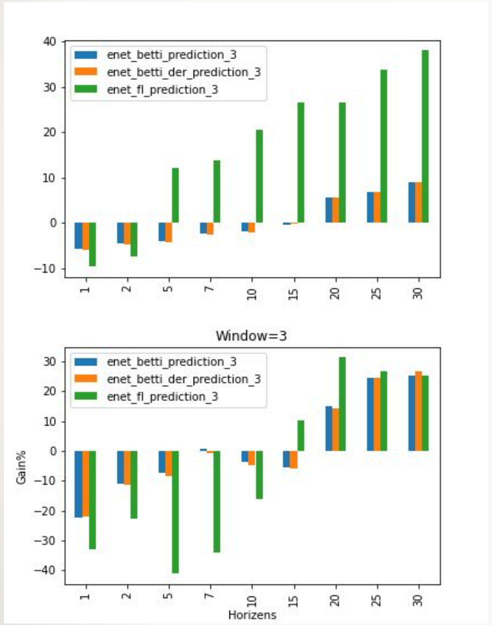
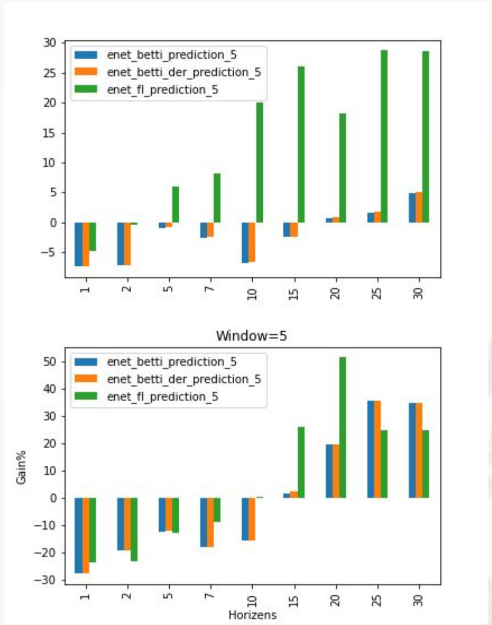


Fig. 20: The figure in the first row is generated by the 2017 Bitcoin transaction graph data, and the figure in the second row is generated by the Bitcoin transaction graph data in 2020. xgbt_betti_prediction_3 represents the XGBT model, the Betti number data set, and the window size is 3. betti_der is the Betti derivative data set, and fl is the filtration data set.

Evaluation

ENET(Bitcoin)





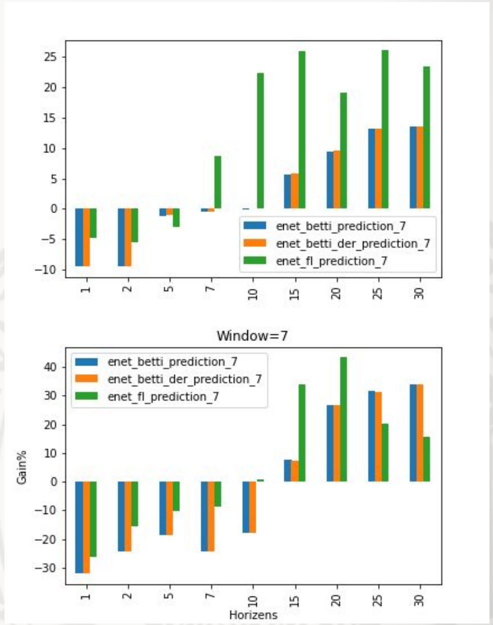


Fig. 21: The figure in the first row is generated by the 2017 Bitcoin transaction graph data, and the figure in the second row is generated by the Bitcoin transaction graph data in 2020. enet_betti_prediction_3 represents the ENET model, the Betti number data set, and the window size is 3. betti_der is the Betti derivative data set, and fl is the filtration data set.

Evaluation

Random Regression(Bitcoin)

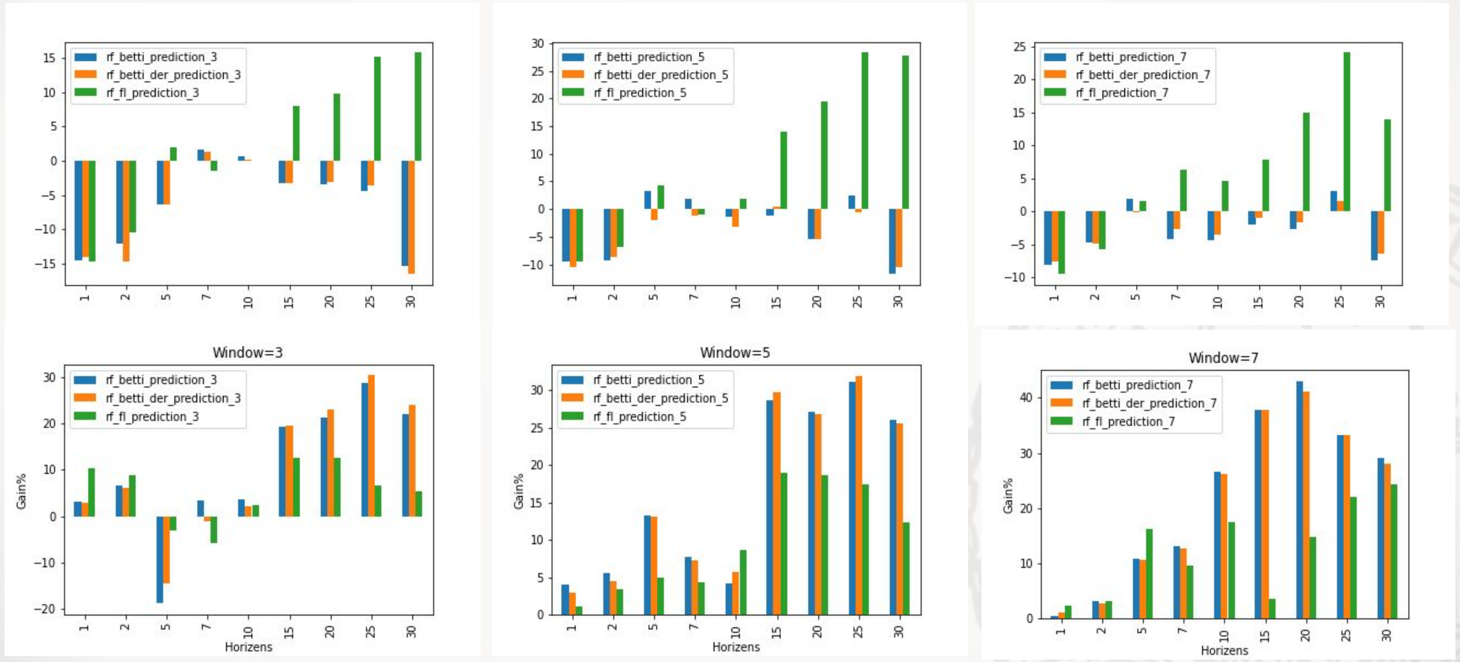


Fig. 22: The figure in the first row is generated by the 2017 Bitcoin transaction graph data, and the figure in the second row is generated by the Bitcoin transaction graph data in 2020. rf_betti_prediction_3 represents the Random Regression model, the Betti number data set, and the window size is 3. betti_der is the Betti derivative data set, and fl is the filtration data set.

Evaluation

Random Regression (Bitcoin + Other cryptocurrencies)

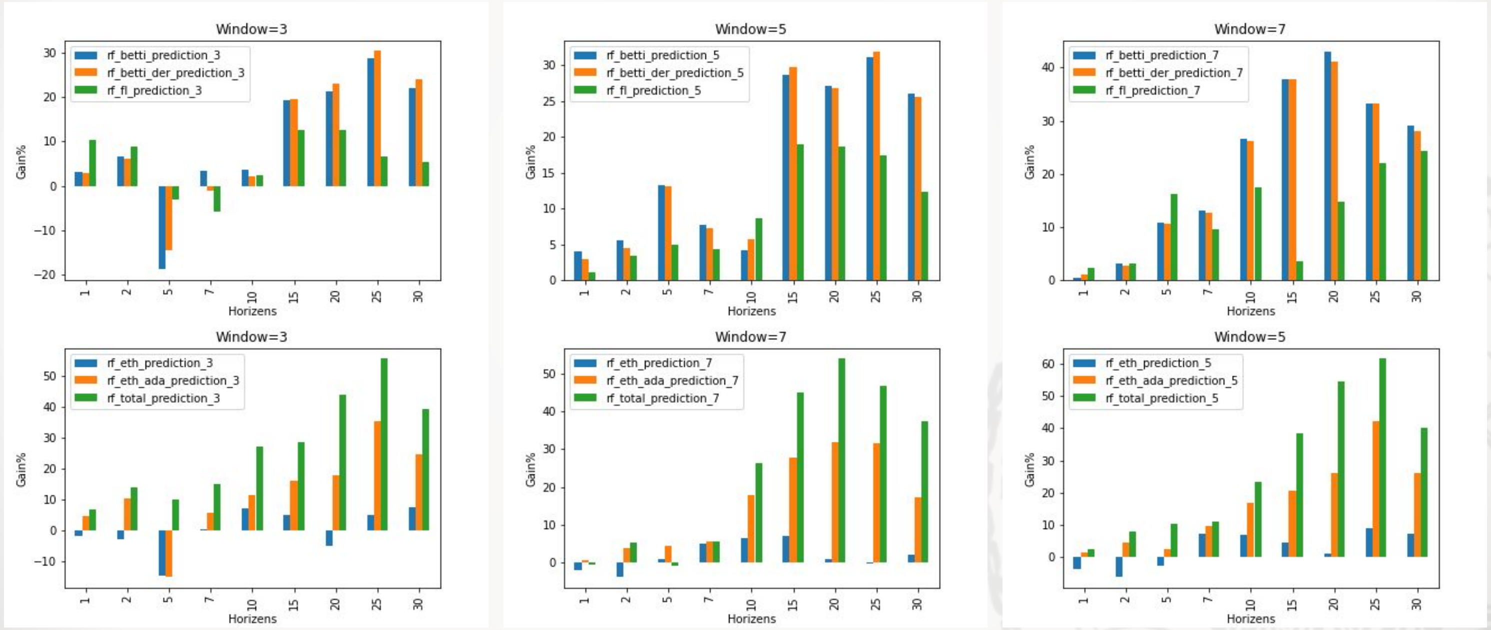


Fig. 23: The figure in the first row is generated only by the Bitcoin transaction graph data, and the figure in the second row is generated by the Bitcoin transaction graph data and other cryptocurrencies data in 2020. ***rf_eth_prediction_3*** represents the Random Regression model, the extra Ethereum transaction data set, and the window size is 3. ***eth_ada*** refers to the extra Ethereum and Cardano data set, and ***total*** is the data set of extra top 7 cryptocurrencies.

Evaluation

Informer

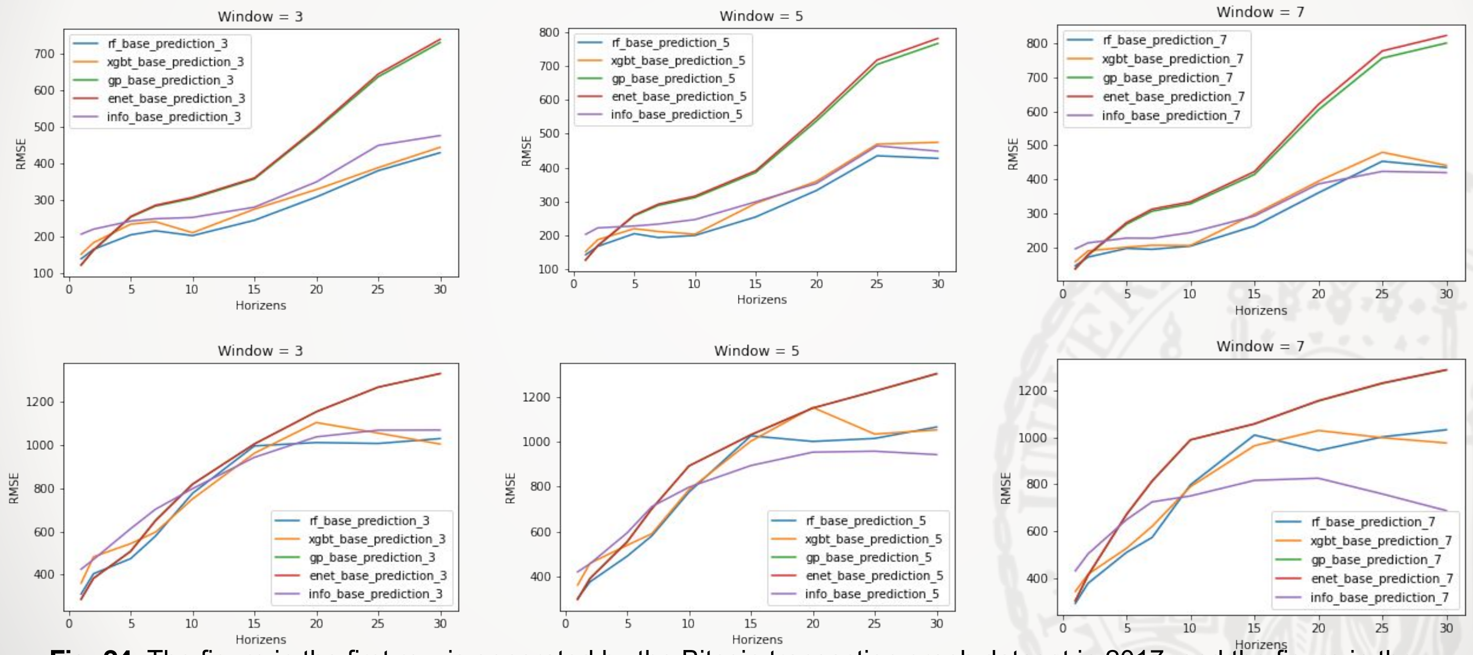


Fig. 24: The figure in the first row is generated by the Bitcoin transaction graph dataset in 2017, and the figure in the second row is generated by the Bitcoin transaction graph dataset in 2020. Here, the base dataset is used on 5 different models, they are random forest(rf), gaussian process(gp), xgbt, enet, informer(info).

Evaluation

ENET(Ethereum)

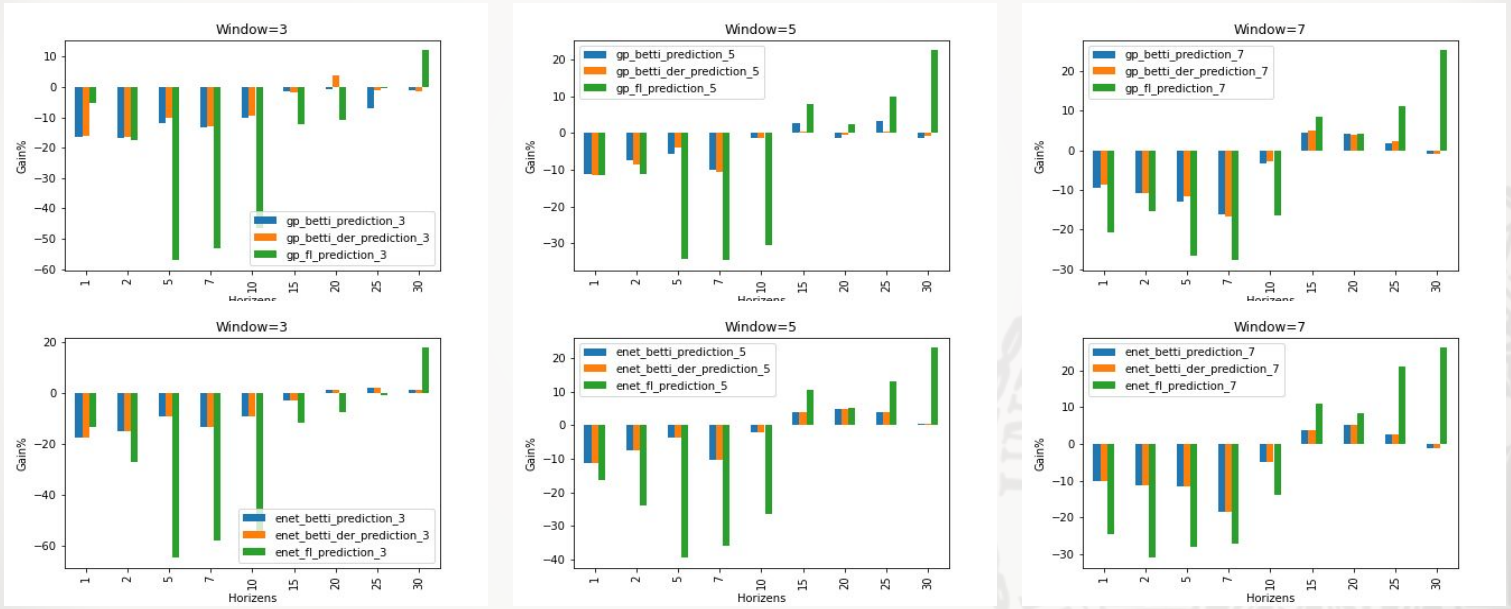
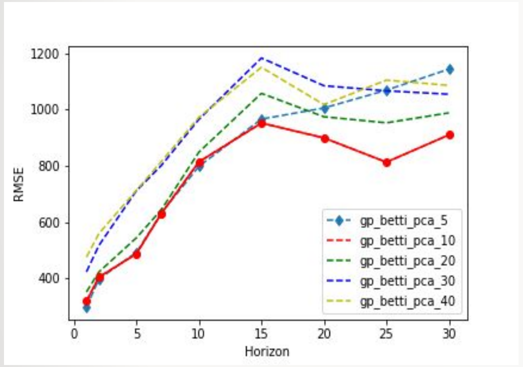
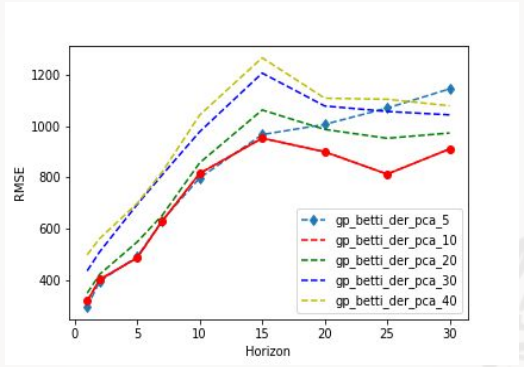


Fig. 25: The figure in the first row is generated by the Ethereum transaction graph data on Gaussian Process model, and the figure in the second row is generated by the Ethereum transaction graph data on ENET model. enet_betti_prediction_3 represents the ENET model, the Betti number data set, and the window size is 3. betti_der is the Betti derivative data set, and fl is the filtration data set.

Evaluation

Principal component analysis(PCA)





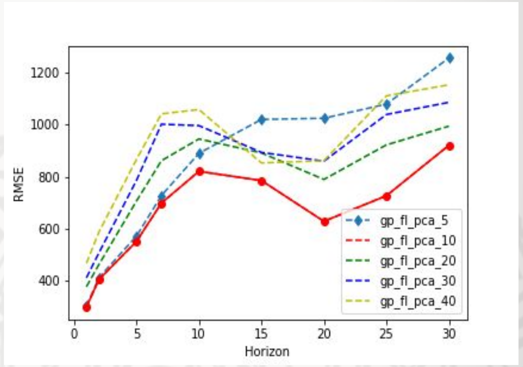


Fig. 26: gp_betti_pca_5 represents the Gaussian Process model, the Betti number data set, and the $n\_components$ of PCA is 5. betti_der is the Betti derivative data set, and fl is the filtration data set.

Conclusion

- The topological features can capture more information in drastically changing graphs.
- The persistent benefit of topological features exists on a wide range of blockchain dynamic graphs(including on Ethereum blockchain).
- In the long-term forecasting scenario, Informer model has better performance.
- Not only internal factors of blockchain have a great influence on cryptocurrency transactions, but external factors (like competitive cryptocurrency) also play an important role.

Reference

[1] Abay, Nazmiye C. ; Akcora, Cuneyt G. ; Gel, Yulia R. ; Islambekov, Umar D. ; Kantarcioglu, Murat ; Tian, Yahui ; Thuraisingham, Bhavani: Chainnet: Learning on blockchain graphs with topological features. In: arXiv preprint arXiv:1908.06971 (2019).

[2] Akcora, Cuneyt G. ; Dey, Asim K. ; Gel, Yulia R. ; Kantarcioglu, Murat: Forecasting bitcoin price with graph chainlets. In: Pacific-Asia conference on knowledge discovery and data mining Springer, 2018, S. 765–776.

[3] Nakamoto, Satoshi: Bitcoin: A peer-to-peer electronic cash system / Manubot. 2019. – Forschungsbericht

[4] Buterin, Vitalik u. a.: Ethereum white paper. In: GitHub repository 1 (2013), S. 22–23.

[5] 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).

[6] Vaswani, Ashish ; Shazeer, 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).

[7] 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.

[8] Rasmussen, Carl E.: Gaussian processes in machine learning. In: Summer school on machine learning Springer, 2003, S. 63–71.

[9] 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

Thank you

Questions?

Definition

Vietoris–Rips complex

In topology, the Vietoris–Rips complex is a way of forming a topological space from distances in a set of points. It is an abstract simplicial complex that can be defined from any metric space M and distance δ by forming a simplex for every finite set of points that has a diameter at most δ. That is, it is a family of finite subsets of M, in which we think of a subset of k points as forming a (k − 1)-dimensional simplex (an edge for two points, a triangle for three points, a tetrahedron for four points, etc.); if a finite set S has the property that the distance between every pair of points in S is at most δ, then we include S as a simplex in the complex.[1]

1. https://en.wikipedia.org/wiki/Vietoris%E2%80%93Rips_complex

Definition

Persistent homology

Persistent homology is a method for computing topological features of a space at different spatial resolutions. More persistent features are detected over a wide range of spatial scales and are deemed more likely to represent true features of the underlying space rather than artifacts of sampling, noise, or particular choice of parameters.

To find the persistent homology of a space, the space must first be represented as a simplicial complex. A distance function on the underlying space corresponds to a filtration of the simplicial complex, that is a nested sequence of increasing subsets. [1]

1. https://en.wikipedia.org/wiki/Persistent_homology

Transformer vs Informer

Disadvantages of transformer:

1. The secondary calculation complexity of self-attention and the operation of the self-attention mechanism will cause the time complexity of our model to be O(L^2);
2. The memory bottleneck of the stacking layer with long input: J encoder/decoder stacks will cause the memory usage to be O(J*L^2);
3. The speed of predicting long output drops sharply: dynamic decoding will cause step-by-step inference to be very slow.

Advantages of Informer :

1. ProbSparse Self-Attention achieves O(LlogL) in time complexity and memory usage, and has considerable performance in sequence-dependent alignment.
2. Self-attention extraction highlights control attention by halving the input of the cascade layer, and effectively handles extremely long input sequences.
3. Although the production decoder is conceptually simple, it predicts a long-term sequence in a forward operation instead of step-by-step, which greatly improves the inference speed of long-sequence prediction.

1. https://en.wikipedia.org/wiki/Persistent_homology

Quantile

1. In statistics and probability, quantiles are cut points dividing the range of a probability distribution into continuous intervals with equal probabilities, or dividing the observations in a sample in the same way.
2. pandas.quantile: pos = 1+(n-1)*p

1. https://en.wikipedia.org/wiki/Persistent_homology