Multimed Tools Appl (2017) 76:20099–20110
DOI 10.1007/s11042-017-4396-4

Behavior pattern clustering in blockchain networks

Butian Huang[1] · Zhenguang Liu[2] · Jianhai Chen[1] · Anan Liu[3] · Qi Liu[2] · Qinming He[1]

Received: 14 November 2016 / Revised: 23 December 2016 / Accepted: 11 January 2017 /
Published online: 26 January 2017

Abstract Blockchain holds promise for being the revolutionary technology, which has the potential to find applications in numerous fields such as digital money, clearing, gambling and product tracing. However, blockchain faces its own problems and challenges. One key problem is to automatically cluster the behavior patterns of all the blockchain nodes into categories. In this paper, we introduce the problem of behavior pattern clustering in blockchain networks and propose a novel algorithm termed BPC for this problem. We evaluate a long list of potential sequence similarity measures, and select a distance that is suitable for the behavior pattern clustering problem. Extensive experiments show that our proposed algorithm is much more effective than the existing methods in terms of clustering accuracy.

Keywords Blockchain technology · Behavior pattern clustering · Clustering · Sequences

✉ Zhenguang Liu
zhenguangliu@zju.edu.cn

Butian Huang
butine@zju.edu.cn

Jianhai Chen
chenjh919@zju.edu.cn

Anan Liu
anan0422@gmai.com

Qi Liu
qiliumize@gmail.com

Qinming He
hqm@zju.edu.cn

[1] Department of Computer Science, Zhejiang University, Hangzhou, China

[2] School of Computing, National University of Singapore, Singapore, Singapore

[3] School of Electronic Information Engineering, Tianjin University, Tianjin, China

1 Introduction

A blockchain is a distributed database that maintains a list of growing records of all the transactions occurred. The growing records are termed blocks. Each block contains a timestamp, a nonce, a reference to (ie. hash of) the previous block and a list of all of the transactions that have taken place since the previous block. Different from traditional centralized databases, blockchain does not have a centralized node. Each node in the blockchain network has an equal right (in query, in sending transactions, and in participating into the consensus process) and maintains a ledger that records every transaction that has ever occurred. For a public blockchain, anyone in the world can join the blockchain network and become a node. For a private chain, only the participation parties that are authorized can become nodes.

Since each node in the blockchain network keeps a ledger of all the transactions occurred, there are multi-backup ledgers in the blockchain network. As such, the vulnerability of a centralized database is eliminated. Furthermore, since all the blocks in the blockchain network can not be modified, blockchain is secured from tampering and artificial modification.

Blockchain holds promise for being the revolutionary technology, which has the potential to find applications in numerous fields. BitCoin has shown how blockchain technologies can enable the creation of a crypto-currency [12, 15]. In more recent times, blockchain applications have appeared that go far beyond their first applition-BitCoin [5, 8, 9, 21]. For instance, blockchain technology has found its applications in domains as varied as transaction processing, government cash management, clearing, medicine tracing, gambling, and personal credit management system [4, 6, 10, 11, 20, 29].

However, blockchain faces its own problems and challenges [23]. In a public blockchain, there are usually millions of nodes. Some nodes may attempt to cheat in the network for illegal interests and have anomalous behavior patterns while the majority nodes behaves normally. It takes a tremendous amount of time and efforts, if possible, to manually study the behaviors of all the nodes. Towards this issue, this paper propose to automatically cluster the behavior patterns of all the nodes into categories. After the clustering, we can select representative behavior patterns for each category as behavior templates. Then utilize the behavior templates to identify strange behavior patterns that do not conform to any template. Moreover, clustering behavior patterns into categories may both leads to deeper insights into the blockchain network and helps maintainers manage and organize the nodes.

This paper seeks to automatically cluster the behavior patterns into categories. The main contributions of this paper can be summarized as below.

- We extract sequences data to represent node behaviors, and propose an algorithm to cluster nodes into categories. To the best of our knowledge, this paper is the first to formulate and address the problem of clustering node behaviors in blockchain networks.
- We conduct extensive experiments to evaluate the effectiveness of our algorithm against the existing methods and study its performances in various settings. Experimental results show that our proposed algorithm termed BPC is much more effective than the existing methods in terms of clustering accuracy.

The rest of the paper is organized as follows. In Section 2, we review the related work. Afterwards, we introduce the formal problem definition of behavior pattern clustering in Section 3. In Section 4, we present the details of our proposed method. Section 5 illustrates the extensive experiments and summarizes the findings. Finally, Section 6 concludes the paper.

2 Related work

In order to cluster the behavior patterns of all the nodes in blockchain network, the first step is to extract features to represent behavior pattern of each node. Since the most important feature of a node is its transaction amount changing over time, we extract the sequences according to the transaction amounts change over time. Each node is represented by a sequence. Our goal is to cluster these sequences into several categories where the number of categories is defined by the user. The most closely related work are sequence similarity measure and clustering approaches.

2.1 Sequence similarity measure

Sequence similarity measure defines the similarity between two sequences being compared. Numerous research in sequences has produced a number of distance measures [2, 14, 19, 22]. The most popular measures are Euclidean distance, Dynamic Time Warping (DTW), Edit Distance on Real sequence (EDR) and Longest Common SubSequences (LCSS).

Euclidean distance requires the two sequences be of the same length. It computes the distance by simply employing the L2 norm. DTW does not require that the two sequences be of the same length and can handle time shifting. To implement this, DTW duplicates the previous elements and calculates an optimal match between the sequences [19]. The LCSS technique [22] introduces a threshold value ϵ to handle noises in sequences, which try to find the longest common subsequences between the given two sequences. EDR [2] leverages gap and mismatch penalties and can handle both noises and time shifting.

2.2 Clustering approaches

Numerous clustering methods have been proposed in the last decades, which can be roughly classified into four major categories: partitioning-based methods, density-based methods, hierarchical methods, and grid-based methods [3, 7, 16–18, 24, 26–28].

Partitioning-based methods divide the n unlabeled data tuples into k partitions. Each partition corresponds to a cluster and has a cluster center. Two renowned heuristic approaches to represent partition based methods are k-means and k-medoids. In k-means, each cluster is represented by the average value of the tuples in the cluster. In contrast, in k-medoids, each cluster is represented by the most centrally located tuple in a cluster.

Hierarchical methods group data tuples into a tree structure [13, 16, 27]. There are generally two kinds of hierarchical methods: agglomerative and divisive. Agglomerative methods start by regarding each tuple as a cluster and then merge clusters into larger and larger clusters, until all tuples are in a single cluster or the desired number of clusters are satisfied. Divisive methods work in the opposite way, where all tuples are started being in the same cluster and then divide the clusters into smaller and smaller clusters. A main problem in hierarchical clustering methods is that any merge or split is unchangeable once executed. Chameleon [16] and BIRCH [27] are two typical hierarchical methods.

Density based methods starts from a cluster of one tuple and keep absorbing neighbor tuples as long as the density (number of tuples within a certain distance) is higher than a threshold. DBSCAN is the most classical method in density based clustering. These methods, however, require to pre-set a few parameters. It is usually difficult to select a good parameter values.

Grid-based methods [1, 25] quantize the feature space into a finite number of cells such that the space forms a grid structure. All the clustering operations are preformed in the

grid structure. A typical example of the grid-based approaches is STING [25]. In STING, several levels of rectangular cells correspond to several different levels of resolution. Statistical information for each cell is pre-computed and stored. Due to the grid structure, these methods have fast computation speeds, the cost is losing a portion of accuracy.

3 Problem statement

The problem of behavior pattern clustering in blockchain networks can be formalized as below.

Definition 1 Given the sequences $\{s_1, s_2, \cdots, s_n\}$ extracted from the n nodes of the blockchain network, and an integer k where k is the number of clusters defined by the user. The goal is to cluster the n sequences into k clusters such that sequences in the same cluster are similar to each other while sequences from different clusters are not similar.

Each node corresponds to a sequence. A sequence is usually extracted as the transaction amount change over time for the node, which is due to the fact that transaction amount is usually the most predominent feature of a node. However, if we are interested in another feature of the node, we can simply extract that feature from the node and use the new sequence instead.

4 Methodology

In this section, we introduce our method to address the problem defined in the previous section (Section 3). In order to cluster the sequences into clusters, the first step is to select a similarity measure for comparing two sequences. We first present a detailed introduction to the similarity measures in Section 4.1 and then elaborate our proposed clustering algorithm termed BPC (Behavior Pattern Clustering) in Section 4.2.

4.1 Similarity measure selection

Sequences similarity measure defines the similarity between two sequences being compared. Numerous sequence similarity measures have been proposed. We will introduce four classical measures, namely Euclidean distance, Dynamic Time Warping (DTW), Edit Distance on Real sequence (EDR) and Longest Common SubSequences (LCSS).

1. **Euclidean distance.** Let s and x be two d-dimensional vectors. The Euclidean distance between x and s is

$$D(s, x) = \sqrt{\sum_{i=1}^{d}(x_i - s_i)^2} \tag{1}$$

2. **DTW distance.** The DTW distance between two sequences x and s is

$$D(s, x) = \begin{cases} 0, & if\ |s| = 0 \quad and \quad |x| = 0 \\ +\infty, & if\ |s| = 0 \quad or \quad |x| = 0 \\ D(s_1, x_1) + min\{D(s - s_1, x), D(s, x - x_1), \\ \quad D(s - s_1, x - x_1)\}, & otherwise \end{cases} \tag{2}$$

where $|s|$ and $|x|$ stand for the length of s and x, respectively. $s - s_1$ means remove s_1 from s. The DTW distance is defined in a recursion way. DTW duplicates the previous elements and calculates distance according to the optimal match between the sequences. We would like to point out that DTW is the most commonly adopted distance.

3. **EDR distance.** The EDR distance between two sequences x and s is

$$D(s,x) = \begin{cases} |x|, & if\ |s| = 0 \\ |s|, & if\ |x| = 0 \\ min\{D(s - s_1, x - x_1) + subcost, & D(s - s_1, x) + 1, \\ \quad D(s, x - x_1) + 1\}, & otherwise \end{cases} \tag{3}$$

where $subcost = 0$ if $|s_{1,x} - x_{1,x}| \leqslant \epsilon$ and $|s_{1,y} - x_{1,y}| \leqslant \epsilon$, and $subcost = 1$ otherwise.

4. **LCSS similarity.** The LCSS similarity between two sequences x and s is

$$C(s,x) = \begin{cases} 0, & if\ |s| = 0 \quad or \quad |x| = 0 \\ C(s - s_1, x - x_1) + 1, & if\ |x_{1,x} - s_{1,x}| \leqslant \epsilon \ \ and \ \ |x_{1,y} - s_{1,y}| \leqslant \epsilon \\ max\{C(s - s_1, x), C(s, x - x_1), & otherwise \end{cases} \tag{4}$$

Different from above three distances, LCSS measures the longest common subsequences between two sequences. LCSS can be seen as a measure of similarity between two sequences rather than a measure of distance.

Since DTW distance is commonly adopted and can deal with two sequences with different lengths, we select DTW distance as the similarity measure between sequences. Note that similarity is inversely proportional to distance. EDR distance and LCCS similarity can also handle two sequences of different lengths. We do not select them as the similarity measure for the following reasons. EDR distance and LCCS similarity are mainly designed to handle the noises in the sequences. However, all the transaction amounts in the blockchain network are precise and no noise is allowed.

4.2 The BPC algorithm for clustering

After selecting the sequence similarity measure, we present our proposed algorithm in Algorithm 1. The BPC algorithm is similar but different to k-means clustering. Lines 1–3 initialize the k cluster centers $o_1, o_2, \cdots, o_k$. Lines 5–6 assign each sequence into an appropriate cluster. Line 7 calculates the new cluster center. Line 8 tests weather the cluster iteration is converged.

There are three main differences between k-means clustering and BPC algorithm. (1) k-means clustering initializes the cluster centers randomly, while BPC sorts the sequences and select k sequences uniformly from the sorted list. (2) k-means clustering utilizes Euclidean distance between static tuples, while BPC utilizes DTW distance between sequences. (3) k-means clustering uses the average value of the tuples in the cluster as the cluster center for that cluster, while BPC selects the sequence with the smallest distance to its $\lceil \frac{n}{k} \rceil$th nearest neighbor among all sequences in the cluster as the cluster center.

Algorithm 1 The BPC (Behavior Pattern Clustering) Algorithm

Input: Sequences $s_1, s_2, \cdots, s_n$; k (number of clusters)
Output: Divide the n input sequences into k clusters

```
Calculate the distance between each pair of sequences according to Equation (2)
Sort the sequences with respect to their distances to their ⌈n/k⌉th nearest neighbors
Select k sequences uniformly from the sorted list as k initial cluster centers
o_1, o_2, ···, o_k
for i ← 1 to max_iter_num do
    for i ← 1 to n do
        Label the cluster of s_i as j if the distance D(s_i, o_j) is the smallest among
        the distances between s_i and all the k cluster centers
    Select each new cluster center o_m(m = 1, 2, ···, k) as the sequence with the
    smallest distance to its ⌈n/k⌉th nearest neighbor among all sequences in the mth
    cluster
    if all the k new cluster centers are the same as the old ones then
        break
return the cluster labels of all sequences
```

5 Experimental evaluation

In this section, we discuss the experiments. First, we evaluate our proposed BPC method against the existing methods in Section 5.1. Then, we study the performance of BPC while replacing our cluster center initialization with random initialization. Finally, we highlight the findings in Section 5.3.

Experimental setup Blockchain enjoys a lot of applications due to that all the finished transactions are transparency to everyone and are not artificially changeable. We conduct experiments on a real blockchain application on stock trading. In the application, there are 1,321 nodes in the blockchain network. The transaction amounts of a node in the recent three months are extracted as a sequence. Our goal is to divide these sequences into several clusters. Leveraging the results, we may identify common behavior templates on transactions, detect the cluster of strange behavior patterns, gain deeper insights into the transaction community, or be able to better organize the nodes.

5.1 Comparison against classical clustering methods

In order to evaluate the performance of BPC on the novel problem of behavior pattern clustering, we compare BPC against the classical DBSCAN and hierarchical clustering (HIC) on the stock trading database. First, we varied the value of k, which is the user defined number of clusters, from 2 to 10. Then we added the cluster label for the 1,321 nodes manually

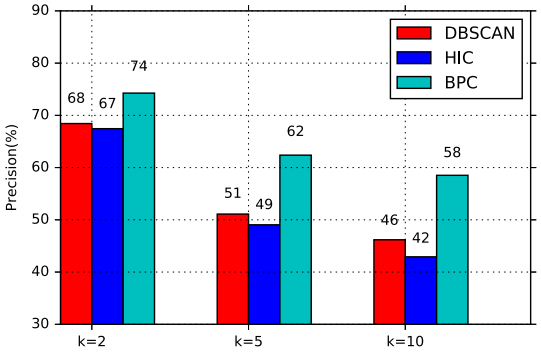
Table 1 Performance comparison with classical methods

Method	Precision		
	$k = 2$	$k = 5$	$k = 10$
DBSCAN (Density Based Method)	68.43 %	51.10 %	46.18 %
HIC (Hierarchical Clustering Method)	67.45 %	49.05 %	42.92 %
BPC	74.26 %	62.38 %	58.52 %

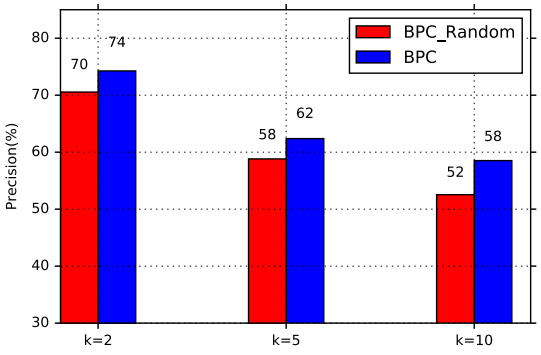
and tested the three clustering methods on the same data set to compare their performances. The performance is measured by the precision of the cluster results according to the ground truth.

Table 1 and Fig. 1a show the performance comparison between the three clustering methods. For the DBSCAN method, the generated number q of clusters can not be guaranteed to be the same as the user defined number k of clusters. If the generated number of clusters of DBSCAN is greater than the user defined k. We keep merging the two clusters with the smallest distance until the number of clusters reduced to k. If the generated number of clusters of DBSCAN is smaller than the user defined k. We re-select the initial cluster centers and re-run the DBSCAN method.

From Table 1 and Fig. 1a, we can clearly see that our proposed BPC method outperforms the classical DBSCAN and hierachical clustering methods in terms of clustering accuracy for the 1,321 sequences. This observation may step from the fact that DBSCAN is very sensitive to the initial cluster centers selected and hierachical clustering methods concentrates on the similarity between two single time series rather than adopting a cluster view. It is worthy to mention that all the environmental settings of the three methods are the same. This further verifies the effectiveness of our proposed method. We can also observe that the precisions of these methods decreases when k increases.



(a) Performance comparison with classical methods.



(b) Comparison with clustering using random cluster center initialization.

Fig. 1 Performance comparison with classical methods and clustering using random cluster center initialization

Table 2 Performance comparison with clustering using random cluster center initialization

Method	Precision		
	$k = 2$	$k = 5$	$k = 10$
BPC_Random	70.55 %	58.82 %	52.54 %
BPC	74.26 %	62.38 %	58.52 %

BPC_Random is short for BPC algorithm with random cluster center initialization

5.2 Studying the effect of removing the proposed cluster center initialization

We also conducted experiments to study the performance of BPC while replacing our cluster center initialization. We replaced our cluster center initialization with random initialization. Then ran the experiments on the same dataset.

The comparison results are demonstrated in Table 2 and Fig. 1b. From the table and the figure, we can observe that the BPC algorithm consistently outperforms the BPC_Random algorithm. This suggests that our proposed cluster center initialization can achieve better performance than random initialization. This dues to the fact that selecting k sequences uniformly from the sorted list makes the clustering process more efficient in capturing the true structures of the clusters. Surprisingly, we find that the BPC_Random algorithm is better than DBSCAN and hierarchical clustering methods. This may due to the fact that DBSCAN and hierarchical clustering need to merge clusters to obtain the required number of clusters.

5.3 Summary of results

In short, our experimental evaluation suggests that: (1) the proposed BPC method outperforms the state-of-the-arts in the behavior pattern clustering task for the 1,321 blockchain nodes; (2) our proposed cluster center initialization can achieve better performance than random initialization; and (3) the precisions of all the tested methods decreases when the user-defined number of clusters increases.

6 Conclusion

In this paper, we have introduced the problem of behavior pattern clustering in blockchain networks and have proposed a novel algorithm termed BPC to address this problem. To the best of our knowledge, this paper is the first to formulate and address the problem of clustering node behaviors in blockchain networks. We have evaluated a long list of potential sequence similarity measures, and selected a distance that is suitable for the behavior pattern clustering problem. We have conducted extensive experiments to evaluate the effectiveness of our algorithm against the existing methods, and studied the effect of its cluster center initialization process. Experimental results show that our proposed algorithm is much more effective than the existing methods in terms of clustering accuracy.

For the future work, we plan to dive further into behavior pattern analysis in blockchain networks in the following two aspects. First, this paper focuses mainly on univariate sequences where each node has a sequence of one variate, however, every node may well have multivariate sequences. Clustering multivariate sequences will be a novel and challenging topic. Second, for multivariate sequences, different variates usually have different importance, how to strengthen important variates while suppress the unimportant ones deserves further study.

References

1. Ankerst M, Breunig MM, Kriegel H, Sander J (1999) OPTICS: ordering points to identify the clustering structure. In: SIGMOD 1999, proceedings ACM SIGMOD international conference on management of data, pp 49–60
2. Chen L, Özsu MT, Oria V (2005) Robust and fast similarity search for moving object trajectories. In: Proceedings of the ACM SIGMOD international conference on management of data, Baltimore, Maryland, USA, June 14–16, 2005, pp 491–502
3. Cheng-Yue R, Hsu R, Stevens N (2012) Discovering elite users in question and answering communities. CS:1–5
4. Christidis K, Devetsikiotis M (2016) Blockchains and smart contracts for the internet of things. IEEE Access 4:2292–2303
5. Croman K, Decker C, Eyal I, Gencer AE, Juels A, Kosba AE, Miller A, Saxena P, Shi E, Sirer EG, Song D, Wattenhofer R (2016) On scaling decentralized blockchains - (a position paper). In: Financial cryptography and data security - FC 2016 international workshops, BITCOIN, VOTING, and WAHC, pp 106–125
6. Dorri A, Kanhere SS, Jurdak R (2016) Blockchain in internet of things: challenges and solutions. arXiv:1608.05187
7. Ester M, Kriegel H, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the second international conference on knowledge discovery and data mining (KDD-96), pp 226–231
8. Forte P, Romano D, Schmid G (2015) Beyond bitcoin - part I: a critical look at blockchain-based systems. IACR Cryptology ePrint Archive 2015:1164
9. Forte P, Romano D, Schmid G (2016) Beyond bitcoin - part II: blockchain-based systems without mining. IACR Cryptology ePrint Archive 2016:747
10. Garay JA (2015) Blockchain-based consensus (keynote). In: 19th international conference on principles of distributed systems, OPODIS 2015, pp 5:1–5:1
11. Gervais A, Karame GO, Wüst K, Glykantzis V, Ritzdorf H, Capkun S (2016) On the security and performance of proof of work blockchains. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, vienna, Austria, October 24–28, 2016, pp 3–16
12. Guadamuz A, Marsden C (2015) Blockchains and bitcoin: regulatory responses to cryptocurrencies. First Monday 20(12)
13. Guha S, Rastogi R, Shim K (2001) Cure: an efficient clustering algorithm for large databases. Inf Syst 26(1):35–58
14. Hirano S, Tsumoto S (2003) Comparison of similarity measures and clustering methods for time-series medical data mining. In: Data mining and knowledge discovery: theory, tools, and technology, pp 219–225
15. Karame G (2016) On the security and scalability of bitcoin's blockchain. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, Vienna, Austria, October 24–28, 2016, pp 1861–1862
16. Karypis G, Han E, Kumar V (1999) Chameleon: hierarchical clustering using dynamic modeling. IEEE Computer 32(8):68–75
17. Liao TW (2005) Clustering of time series data: a survey. Pattern Recogn 38(11):1857–1874
18. Liu L, Zhu F, Jiang M, Han J, Sun L, Yang S (2012) Mining diversity on social media networks. Multimedia Tools and Applications:179–205
19. Morse MD, Patel JM (2007) An efficient and accurate method for evaluating time series similarity. In: Proceedings of the ACM SIGMOD international conference on management of data, Beijing, China, June 12–14, 2007, pp 569–580
20. Petersz GW, Panayiy E (2015) Understanding modern banking ledgers through blockchain technologies: future of transaction processing and smart contracts on the internet of money. SSRN 2692487(1):1–33
21. Underwood S (2016) Blockchain beyond bitcoin. Commun ACM 59(11):15–17
22. Vlachos M, Gunopulos D, Kollios G (2002) Discovering similar multidimensional trajectories. In: Proceedings of the 18th international conference on data engineering, pp 673–684
23. Vukolic M (2015) The quest for scalable blockchain fabric: proof-of-work vs. BFT replication. In: Open problems in network security - IFIP WG 11.4 international workshop, inetsec 2015, pp 112–125
24. Wang F, Chen W, Wu F, Zhao Y, Hong H, Gu T, Wang L, Liang R, Bao H (2014) A visual reasoning approach for data-driven transport assessment on urban roads. In: 2014 IEEE conference on visual analytics science and technology, VAST 2014, Paris, France, October 25–31, 2014, pp 103–112
25. Wang W, Yang J, Muntz RR (1997) STING: A statistical information grid approach to spatial data mining. In: VLDB'97, proceedings of 23rd international conference on very large data bases, pp 186–195
26. Xu R, Wunsch II DC (2005) Survey of clustering algorithms. IEEE Trans Neural Networks 16(3):645–678

27. Zhang T, Ramakrishnan R, Livny M (1996) BIRCH: An efficient data clustering method for very large databases. In: Proceedings of the 1996 ACM SIGMOD international conference on management of data, pp 103–114
28. Zhou X, Wang W, Jin Q (2015) Multi-dimensional attributes and measures for dynamical user profiling in social networking environments. Multimedia Tools and Applications:5015–5028
29. Zyskind G, Nathan O, Pentland A (2015) Decentralizing privacy: using blockchain to protect personal data. In: 2015 IEEE symposium on security and privacy workshops, SPW 2015, pp 180–184

Butian Huang is currently a PhD candidate with the Department of Computer Science, Zhejiang University, China. He received the Master degree from University of Electronic Science and Technology of China in Computer Science. His research interests include data mining, anomaly detection, and multimedia mining.

Zhenguang Liu is currently a postdoctoral research fellow with the School of Computing, National University of Singapore, Singapore. He received the Ph.D. degree from Zhejiang University, and the BS degree from Shandong University, all in Computer Science. His research interests include data mining, anomaly detection, and multimedia mining.

Jianhai Chen is currently a lecturer with the Department of Computer Science, Zhejiang University, China. He received the Ph.D. degree from Zhejiang University, and the BS degree from Hunan University. His research interests vinclude virtualization, high performance computing, and multimedia mining.

Anan Liu is currently an associate professor with the Department of Computer Science, Zhejiang University, China. He received the Ph.D. degree from Tianjin University, and the BS degree from Tianjin University. His research interests include biomedical image processing, learning-based computer vision, and multimodel-based multimedia Mining.

Qi Liu is currently a PhD student in the School of Computing, National University of Singapore, Singapore. He received the Master degree from National University of Singapore, and the BS degree from Shandong University, all in Computer Science. His research interests include data mining, deep learning, and multimedia mining.

Qinming He is currently a professor with the Department of Computer Science, Zhejiang University, China. He received the Ph.D. degree from Zhejiang University, and the BS degree from Zhejiang University, all in computer science. His research interests include virtualization, data mining, and Multimedia Mining.