

Characterizing and Utilizing the Interplay Between Core and Truss Decompositions

Penghang Liu
 University at Buffalo
 Buffalo, USA
 penghang@buffalo.edu

Ahmet Erdem Sariyüce
 University at Buffalo
 Buffalo, USA
 erdem@buffalo.edu

Abstract—Finding the dense regions in a graph is an important problem in network analysis. Core decomposition and truss decomposition address this problem from two different perspectives. The former is a vertex-driven approach that assigns density indicators for vertices whereas the latter is an edge-driven technique that put density quantifiers on edges. Despite the algorithmic similarity between these two approaches, it is not clear how core and truss decompositions in a network are related. In this work, we introduce the vertex interplay (VI) and edge interplay (EI) plots to characterize the interplay between core and truss decompositions. Based on our observations, we devise CORE-TRUSSDD, an anomaly detection algorithm to identify the discrepancies between core and truss decompositions. We analyze a large and diverse set of real-world networks, and demonstrate how our approaches can be effective tools to characterize the patterns and anomalies in the networks. Through VI and EI plots, we observe distinct behaviors for graphs from different domains, and identify two anomalous behaviors driven by specific real-world structures. Our algorithm provides an efficient solution to retrieve the outliers in the networks, which correspond to the two anomalous behaviors. We believe that investigating the interplay between core and truss decompositions is important and can yield surprising insights regarding the dense subgraph structure of real-world networks.

Index Terms—dense subgraph discovery, k -core decomposition, k -truss decomposition

I. INTRODUCTION

Dense subgraphs in real-world networks contain significant and unusual information. There are many application domains where dense subgraphs are useful. A few use cases are finding price value motifs in financial networks [1], locating spam link farms in webs [2], and detecting DNA motifs in biological networks [3]. Core and truss decompositions are effective models to find dense regions with hierarchical relationships. In the core decomposition [4], vertices are assigned density pointers, core numbers, that indicate the cohesiveness in the neighborhood. Truss decomposition [5] yields truss numbers for edges which can be interpreted as the link strength.

Given the wide application space, core and truss decompositions are unified and extended by new models for different types of graphs [6], [7]. However, the relationship between the core and truss decompositions has been overlooked, and it is not clear what aspects of the graph structure are covered

by each decomposition. Understanding the interplay between those measures can enable more effective network analysis.

In this work, we investigate the interplay between core and truss decompositions in real-world networks and random graphs. Our first contribution is the vertex interplay (VI) and edge interplay (EI) plots, which analyze the structure of dense subgraphs from two different perspectives. Then, we propose the CORE-TRUSSDD algorithm to identify the anomalies with respect to the core-truss interplay. We use several real-world networks from various domains and examine the similarities and differences in the core-truss interplay. Our analysis on VI and EI plots gives interesting results for the core-truss interplay behavior in real-world networks. We also show that those behaviors cannot be captured when the edges are rewired with a realistic random graph model. Our algorithm provides an efficient solution to retrieve the outliers in the networks, which reveal the anomalous behaviors observed in the VI plots. Codes for reproducing the results and figures are available at <https://github.com/penghangliu/Core-Truss>.

II. BACKGROUND

Our study explores real-world networks which can be represented as a simple undirected unweighted graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. We represent the neighborhood of a vertex u as $N(u)$.

A. Core decomposition

The k -core subgraph is introduced by Seidman [4] for social networks analysis. **The k -core is a connected, maximal subgraph such that every node in the subgraph has degree of at least k within the subgraph.** The core number of a node

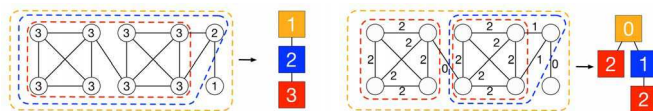


Fig. 1: Examples for core (left) and truss (right) decompositions. On the left, core numbers are shown for each node and red, blue, and orange regions show the 3-, 2-, and 1-cores. They form a hierarchy by containment as denoted. For the same graph, trusses and truss numbers of edges are presented on the right. The entire graph is a 0-truss and the five nodes on the right form a 1-truss. There are two 2-trusses and one of them is a subset of the 1-truss, as denoted by the tree hierarchy.

u , denoted as $K(u)$, is the highest value of k such that the node belongs to a k -core. The maximum core number of all vertices in the graph is defined as the (*core*) *degeneracy*. Core decomposition is the process of finding the core numbers of nodes. This can be achieved by an efficient iterative peeling algorithm in $O(|E|)$ time [8]. k -core subgraph (for any k) is found by a traversal that visits all reachable nodes whose core numbers are at least k . The nested structure of k -cores reveals a hierarchy by containment. Fig. 1 (left) presents an example for the core numbers, k -core subgraphs, and hierarchy.

B. Truss Decomposition

The k -truss subgraph is inspired by the k -core. It considers the edges and the triangles they participate in [9]. **k -truss is a connected, maximal subgraph such that every edge in the subgraph participates in at least k triangles within the subgraph.** The truss number of an edge (u, v) , denoted as $T(u, v)$, is the highest k such that the edge is part of a k -truss. The maximum truss number of all edges in the graph is defined as the *truss degeneracy*. Similar to the core decomposition, the truss decomposition can be achieved by a peeling algorithm in $O(\sum_{v \in V} d(v)^2)$ time. k -trusses also exhibit a hierarchy by containment. Fig. 1 (right) presents the truss numbers of edges, k -truss subgraphs, and their hierarchy on a toy graph.

Relationship between k -clique, and k -core, k -truss. The k -clique is a fully connected subgraph that contains k nodes. For any edge (u, v) in a k -clique, both u and v are connected to the other $k - 2$ nodes, thus every edge in the k -clique participates in at least $k - 2$ triangles within the clique. This shows that k -clique is also a $(k - 1)$ -core and a $(k - 2)$ -truss.

C. Random Graph Models

Random graph models are commonly used as the null model for analyzing real-world networks. The Block Two-Level Erdős-Rényi (BTER) model [10] can simulate networks with community structures by preserving the degree distribution and the clustering coefficient per degree distribution to the best extent. A graph is generated through two phases in the BTER model. In the first phase, vertices are clustered into communities and ER model is applied to generate edges within the same community. In the second phase, edges between communities are generated regarding the size of the communities.

III. DATASETS

In order to explore patterns in various types of real-world networks, we cover datasets from five different categories: social, autonomous systems, citation, collaboration, and web hyperlink networks. The datasets are obtained from SNAP [11], DBLP [12], and Konect [13]. Various statistics are summarized in Table I. For collaboration networks we consider three co-authorship networks from DBLP, which include the venues of data mining, database, and parallel processing respectively. We also consider the random graph models to validate our findings in real-world networks. Table I shows the average core and truss degeneracy of 10 random graphs generated by BTER model for each real-world network.

IV. VERTEX BASED ANALYSIS

In this section, we analyze the interplay of core and truss numbers from the perspective of a vertex. We introduce the **vertex interplay (VI) plot** to demonstrate the spectrum of edges around vertices with a particular core number. Fig. 2 presents the VI plots for some real-world networks in our dataset (all are available in [14]). We examine the neighborhood of vertices with a particular core number and consider the maximum and minimum truss numbers of edges adjacent to a given vertex for a fixed core number. If there are multiple vertices with the same core number, we report the average and interquartile ranges. Formally, for each core number c , we find the set $S = \{u \in V : K(u) = c\}$ and compute $\frac{1}{|S|} \sum_{u \in S} \min\{T(u, v) : v \in N(u)\}$ (likewise for maximum) along with the interquartile ranges.

One general behavior we observe is that the maximum truss number of adjacent edges is strongly correlated to the core number of the vertex; larger core numbers yield larger truss numbers overall. Figure 2a shows this pattern in the VI plots for Catster, which is also commonly observed in other social networks. Regarding the minimum truss numbers, on the other hand, there is a consistent trend for all the vertices regardless of their core numbers. Every vertex in the network is connected to at least one edge with a very low truss number. This is in line with the core-periphery structure [15]; each vertex in the core block is connected to both core and periphery blocks – truss numbers of the edges between core

TABLE I: Statistics of real-world networks and random graphs. The last four columns show the core degeneracy and truss degeneracy numbers. In each group, *Exact* shows the core and truss degeneracy numbers of real-world graphs, and *BTER* presents the same numbers (on average) for the random graphs generated with the BTER model.

Category	Name	V	E	Core degen.		Truss degen.	
				<i>Exact</i>	<i>BTER</i>	<i>Exact</i>	<i>BTER</i>
Social	Catster	150K	5.45M	419	312	205	164
	Dogster	427K	8.54M	248	300	91	192
	Email	36.7K	184K	43	47	20	27
	Email-Eu.	1.00K	25.5K	34	36	21	11
	Flickr	1.72M	15.6M	568	310	276	110
	LiveJour.	4.00M	34.7M	360	33	350	6
	Orkut	3.07M	117M	253	80	76	40
YouTube	1.13M	2.99M	51	78	17	48	
Auto. sys.	As-733	6.47K	12.6K	12	13	8	11
	Caida	26.5K	53.4K	22	28	14	23
	Gnutella	62.6K	148K	6	6	2	1
	Oregon-2	10.9K	31.2K	31	22	23	16
Skitter	1.70M	11.1M	111	191	66	146	
Citation	CiteSeer	384K	1.74M	15	14	11	3
	Cora	23.2K	89.2K	13	9	9	3
	DBLP	12.6K	49.6K	12	13	7	5
	HepTh	27.7K	352K	37	31	28	14
	Patent	3.78M	16.5M	64	9	34	1
Collab.	DBLP_dbs	8.10K	23.0K	35	10	34	2
	DBLP_dm	16.4K	33.9K	24	7	23	1
	DBLP_pp	8.41K	22.9K	44	16	43	3
Web	BerkStan	685K	6.65M	201	258	199	178
	Blogs	1.22K	16.7K	36	38	23	12
	Google	876K	4.32M	44	86	42	72
	NotreDame	326K	1.09M	155	144	153	47
	Stanford	282K	1.99M	71	123	60	93

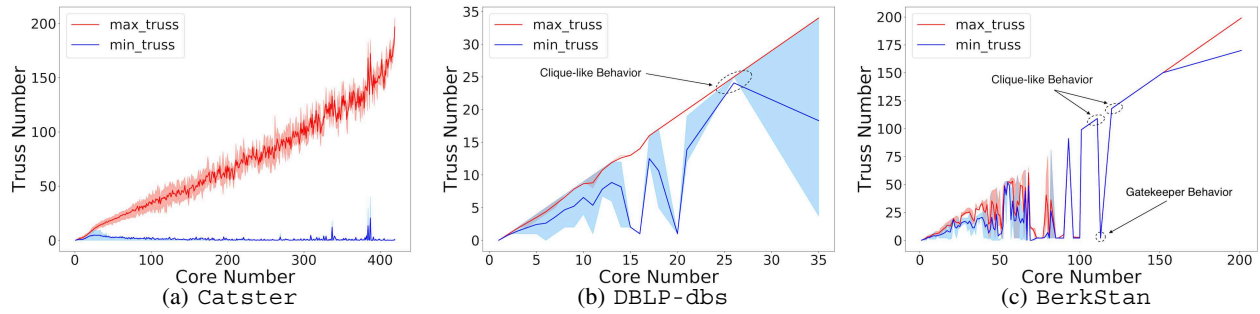


Fig. 2: Vertex interplay (VI) plots for some real-world graphs (all are available in [14]). For each vertex with a particular core number, the maximum and minimum of the truss numbers of surrounding edges are shown. Average and interquartile ranges are computed over the vertices with the same core number.

and periphery are likely to be low. Note that the variation for vertices with the same core number is also very small, i.e., interquartile ranges are narrow, suggesting a high similarity among those vertices. We observe this behavior in 7 (of 8) social networks and most autonomous systems.

Collaboration networks exhibit a different behavior in the VI plots. Figure 2b presents the DBLP-dbs network. The minimum truss numbers are very close to the maximum ones, as opposed to the consistent trend in the social networks. Vertices with large core numbers are not connected to any edge with a low truss number. This implies that cliques of vertices with large core numbers are surrounded by some other cliques with close core numbers. In some cases the minimum truss number, maximum truss number, and core number are almost equal, indicating both k -core and k -truss are derived from a clique-like structure (Fig. 3a), as annotated in Fig. 2b. In fact, a collaboration network can be seen as a union of cliques, where authors of each paper form a clique, which in turn yields closed clique-like structures that are not connected to the nodes in the periphery.

We observe yet another distinct behavior in web networks. Figure 2c shows the VI plot for BerkStan network. The maximum truss numbers are very small for some vertices with large core numbers, implying that the truss numbers of all adjacent edges are small. Most of the neighbors are isolated from each other, indicating a sparse neighborhood despite the cohesiveness suggested by the large core number. Those vertices serve as structural holes in the network [16], where

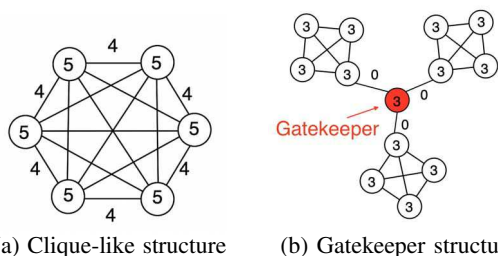


Fig. 3: Examples of clique-like and gatekeeper structure behaviors. In the 6-clique, the core numbers (5) and the truss numbers (4) are close. The gatekeeper (red) has three neighbors isolated from each other. Although the core number (3) shows that the gatekeeper belongs to a cohesive subgraph, its neighborhood structure is not cohesive, as truss numbers are zero.

a node with a large core number is connecting multiple cores isolated from each other (Fig. 3b). In Fig. 2c, there are 271 vertices with core number 113 but the largest truss number adjacent to those is only 2. Note that, in the same network, the vertices with core number 111 and 120 are connected to very large truss numbers, indicating a clique-like structure.

Remark. We also investigate the VI plots of random graphs [14] generated by BTER model, which capture the common pattern observed in social networks but fail to capture the other interesting behaviors in collaboration and web graphs. This shows that the interesting behaviors driven by the unusual structures (cliques and structural holes), seem to be distinctive characteristics in real-world graphs. Given the benefit of analyzing the interplay of core and truss numbers, it is natural to think about using just the degrees of vertices and triangle counts of edges for a similar analysis. However, the skewed distributions of degrees/triangle counts prevent such analyses; the VI plots become inconsistent for the networks from the same domain and behaviors can be gamed with simple changes in the graph. In the extended version [14], we present a few examples. In all variations (degree-triangle count, degree-truss, or core-triangle count) the VI plots fail to generate a consistent and pervasive pattern. **Core and truss numbers can be considered as the regularized and more robust versions of the vertex degrees and edges' triangle counts.**

Summary. VI plots present a meaningful graph summary by showing the interplay between core and truss numbers. Considering the cohesiveness around vertices (i.e., core numbers) with respect to various neighborhoods they are involved in (i.e., truss numbers) is an effective way to understand the dense regions and structural holes. VI plots of networks belonging to the same domain exhibit consistent behaviors whereas the ones from different domains suggest diverse characteristics. We believe that VI plots would be handy for domain practitioners to analyze the network structure.

V. EDGE BASED ANALYSIS

In this section we introduce the **edge interplay (EI) plot** to address the interplay from the perspective of an edge. Figure 4 presents the EI plots for some real-world networks in our dataset (all are available in [14]). Here we examine the truss number of edges between two vertices with particular core numbers. If there are multiple edges having the same pair of core numbers for their endpoints, we show the mean

value of the truss numbers. Formally, for each pair of core numbers $c_1 \leq c_2$, we find the set $S = \{(u, v) \in E : K(u) = c_1, K(v) = c_2\}$ and compute $\frac{1}{|S|} \sum_{(u,v) \in S} T(u, v)$. In addition to the core-truss interplay, we consider the vertex degrees and edges' triangle counts for a similar analysis. For all EI plot variations (degree-truss, degree-triangle count, and core-triangle), we observe consistent behaviors in networks from the same domains. Those alternative EI plots allow us to discover some interesting behaviors.

One general behavior we observe is that the truss number of an edge is strongly correlated to the core numbers of the two endpoints. This also holds true for all the EI plot variations regarding vertex degrees and edge triangle counts. Figure 4a shows this pattern in the EI plot of Email network, and we observe this pattern in all social networks, autonomous systems, and collaboration networks.

Citation networks present a different behavior in the EI plots. Figure 4b presents the EI plot of HepTh network, where edges with large truss number are connecting two endpoints with non-maximum degrees. Another interesting behavior appears consistently in web networks. Figure 4c presents the EI plot of BerkStan network. Edges with large truss number are connecting two vertices with very different degrees. This implies that there are cohesive structures connecting core and periphery blocks. This structure seems to be artificially constructed for special purposes, e.g., spam link farms.

Summary. Similar to the VI plots, the EI plots present a meaningful graph summary by showing the interplay from the edge perspective. We also checked the EI plots of random graphs generated by BTER model (in [14]) and show that the interesting behaviors we observed in EI plot of web and citation networks are, again, distinctive characteristics of real-world networks driven by specific network structures. In addition to the core and truss interplay, the EI plots are also capable to capture consistent behavior in the interplay between degrees and triangle counts. This allows us to distinguish diverse characteristics from more domains.

VI. ANOMALY DETECTION BY CORE-TRUSS DISCREPANCY

Based on our observations in Section IV and Section V, we design the CORE-TRUSS DISCREPANCY DETECTION al-

Algorithm 1 CORE-TRUSSDD

Input: $G = (V, E)$
Output: The set A of anomalous vertices

- 1: $A \leftarrow \emptyset, \mathcal{P} \leftarrow \emptyset$
- 2: $K \leftarrow \text{CORE-DECOMPOSITION}(G)$
- 3: $T \leftarrow \text{TRUSS-DECOMPOSITION}(G)$
- 4: $\text{threshold} \leftarrow \text{sort}(K)[\frac{|V|}{4}]$ \triangleright filter nodes by core numbers
- 5: **for all** $v \in V$ **do**
- 6: **if** $K[v] \geq \text{threshold}$ **then**
- 7: $\vec{P}_v \leftarrow \text{truss-profile of } v$ \triangleright based on T
- 8: $\mathcal{P}.\text{push}(\vec{P}_v)$
- 9: $[C_1, C_2, \dots, C_k] \leftarrow \text{k-means}(\mathcal{P}, k)$ \triangleright vertex clustering
- 10: **for** $C_i \in [C_1, C_2, \dots, C_k]$ **do**
- 11: **for** $v' \in C_i$ **do**
- 12: $z[v'] \leftarrow \text{Z-score of } K[v'] \text{ in cluster } C_i$
- 13: **if** $|z[v']| > 2$ **then**
- 14: $A.\text{push}(v')$
- 15: **return** A

gorithm (CORE-TRUSSDD) to detect the vertices showing anomalous behaviors of core-truss interplay, and apply our proposed algorithm on Email-Eu-core and BerkStan.

A. CORE-TRUSSDD algorithm

Algorithm 1 provides the pseudocode of CORE-TRUSSDD. We first run the core and truss decompositions (Line 2 and 3) to compute the core numbers of all vertices and the truss numbers of all edges. We introduce the vertex truss-profile to represent the spectrum of truss numbers of all adjacent edges for a vertex (Lines 5 to 8).

Definition 1 (Vertex Truss-profile): Given a graph $G = (V, E)$ and its truss degeneracy $\text{max}(T)$, the truss-profile of vertex $v \in V$ is a vector $P_v = [p_0, p_1, \dots, p_{\text{max}(T)}]$, where $p_i = \frac{n(i)}{\sum_{i=0}^{\text{max}(T)} n(i)}$ and $n(i)$ is the number of adjacent edges having truss number i .

Real-world networks often present heterogeneous structures. It is hard to capture a general pattern in truss-profiles of all vertices as they can be highly diverse, making anomaly detection challenging. To address this issue, we group the vertices with similar truss-profiles (Line 9). Here we apply the k-means clustering [17], which minimizes the within-cluster squared distances. By default each vertex will be assigned to a cluster, including those in the periphery with low importance.

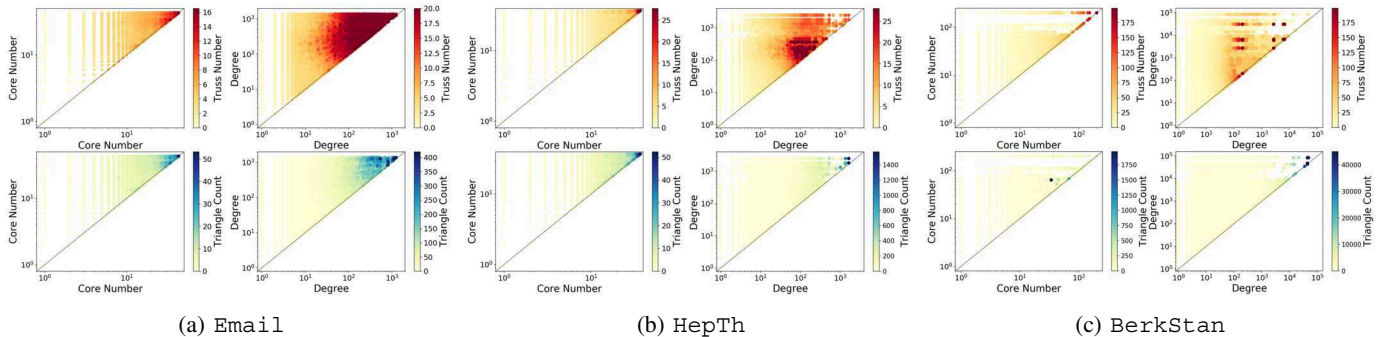


Fig. 4: Edge interplay (EI) plots for some real-world graphs. For each pair of endpoints with particular core numbers/degrees, the average of the truss numbers/triangle counts of the edges are shown.

In order to prevent the false positives caused by these vertices, we ignore the 25% of nodes with the lowest core numbers (Lines 4 and 6).

After clustering the vertices with respect to the truss-profiles, we investigate the patterns and anomalies of core numbers in each cluster. For this purpose, we compute the Z-score of each vertex in order to measure the deviation of their core numbers with respect to the general patterns (Line 12). For a given vertex v with core number K_v , the Z-score is calculated as $z_v = \frac{K_v - \mu}{\sigma}$, where μ and σ are the mean and standard deviation of all core numbers in the cluster. In Lines 13 and 14, we identify the outliers which have the absolute values of Z-scores larger than 2. This corresponds to the 2.28% of the population in normal distribution. We also plot the distribution of core numbers and identify anomalies which deviate from the general patterns. Note that the k-means algorithm only serves as a subroutine in our anomaly detection algorithm. The purpose of clustering is to partition the vertices into several groups where we can expect common truss-profile behaviors. The clustering performance is not our major concern as we only aim to have a scaled-down problem size for the anomaly detection.

Time complexity: CORE-TRUSSDD starts with core and truss decomposition, which takes $O(|E|)$ and $O(\sum_{v \in V} d(v)^2)$ time respectively. The algorithm generates truss-profiles for all vertices in $O(|V| \max(T)) = O(|V|)$ time, since the truss degeneracy $\max(T)$ is $O(1)$ for real-world networks. The k-means clustering can be implemented by the Lloyd's algorithm [18], which takes $O(k|V| \max(T) 2^{\sqrt{|V|}})$ time, where k is the number of clusters. Therefore, the overall time complexity is $O(\sum_{v \in V} d(v)^2 + k|V| 2^{\sqrt{|V|}})$.

Space complexity: We first construct K and T for core and truss decompositions, which take $O(|E|)$ space. In addition, we need $O(|V| \max(T)) = O(|V|)$ space for the truss-profiles \mathcal{P} . The k-means algorithm need $O((|V| + k) \max(T)) = O(|V|)$ space. Arrays A and $[C_1, C_2, \dots, C_k]$ take $O(|V|)$ space, so the total space complexity is $O(|E|)$.

B. Anomalies in real-world networks

In general, we observe similar core numbers within the clusters. Within the cluster, the major population of core

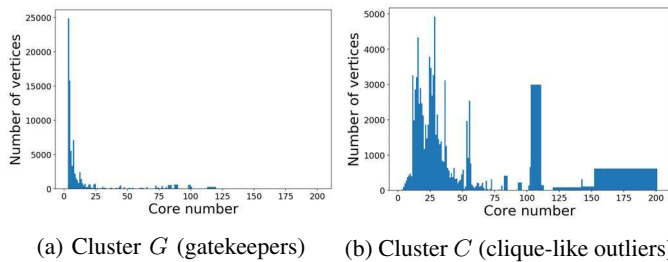


Fig. 5: Core number distributions of anomalous clusters in BerkStan. Each figure represents the core number distribution within a cluster. Fig. 5a shows an anomalous cluster where we observe outliers with core numbers deviated from the major population ($K > 50$). These outliers are found to be the gatekeepers in later analysis. We also observe anomalous vertices in Fig. 5b ($K > 150$), which appear to be clique-like structures.

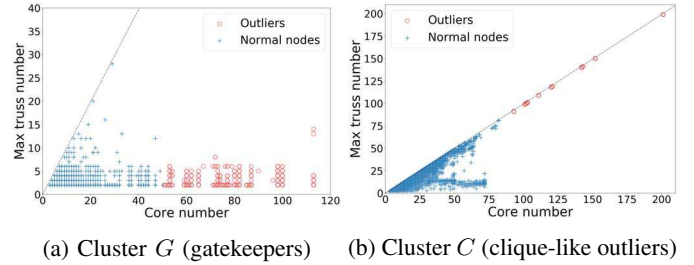


Fig. 6: Gatekeeper and clique-like anomalies in BerkStan clusters. In each scatter plot, the blue crosses denote normal nodes and the red circles represent outliers. The x-axis represents the core number of the vertex, while the y-axis denotes the maximum truss number of the adjacent edges. The dashed line is $y = x - 1$, which indicates the core-truss behavior of cliques.

numbers lies in a small interval. Since the vertices are clustered based on their truss-profiles, it indicates that the core and truss decompositions are consistent. We observe the consistent patterns in 13 out of 15 clusters in BerkStan, as well as in the most clusters in Email-Eu-core. In the remaining two clusters (clusters G and C) of BerkStan, we identify significant amounts of outliers (more than 5%). Fig. 5a shows the core number distribution in cluster G . The majority of vertices have core numbers ranging from 0 to 25, while some outliers have much larger core numbers. Fig. 6a shows that the core numbers of these outliers are much greater than the maximum truss numbers of their adjacent edges. This indicates that the anomalies discovered in this cluster are the gatekeepers illustrated in Fig. 3b. In Fig. 5b we observe that most vertices in cluster C have their core numbers lie between 0 and 55. We identify 5352 outlier vertices with core number larger than 92. As shown in Fig. 6b, the core numbers are extremely close to the maximum truss numbers of the adjacent edges. For all of the 5352 outliers, the difference between their core numbers and the maximum truss numbers is equal to 2. According to our discussion in Section IV, these are the clique-like structures (Fig. 3a) in the network.

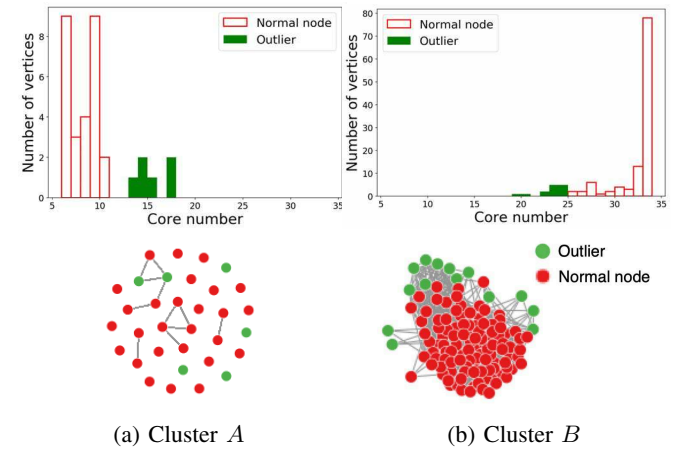


Fig. 7: Core number distributions and the corresponding subgraphs of Email-Eu-core clusters. The histograms present the core number distributions within the clusters. The subgraphs at the bottom are induced by the vertices in the clusters, with normal nodes colored in red and outliers colored in green.

Next, we investigate the patterns and anomalies in the Email-Eu-core clusters. Among the 15 clusters, four of them present anomalous behaviors. Due to the space limit we only show two of them in Fig. 7, along with the core number distributions. In cluster *A* we observe the majority of vertices have core numbers from 6 to 10, while there are six outliers with core numbers larger than 12. Similar anomalies are observed in the other two clusters, where the outliers have core numbers significantly larger than the majority of the population in the cluster. Cluster *B* presents a different picture, where most of the vertices have core numbers of 33 and the outliers have core numbers less than 25. Fig. 7 shows the subgraphs induced by the vertices in the clusters. Note that we are not doing graph clustering but cluster the vertices with respect to the truss-profiles, so they may not be connected at all. We noticed that the subgraph induced by clusters *A* is structurally sparse. Moreover, outliers are less likely to be connected with each other than the normal vertices. The subgraph of cluster *B* shows a very different picture, where the normal vertices form a connected component with relatively high density. This is due to the overall high core numbers in the cluster. Note that the outliers lie in the periphery of the subgraph and few connections exist between them. .

VII. RELATED WORK

The core and truss decompositions, proposed by [4] and [9], is commonly used in dense subgraph discovery. One of the works that addresses the relation between core and truss decompositions is [6], [7]. Sariyuce et al. proposed a network decomposition framework which addresses both core and truss decompositions, and explored the hierarchical and overlap relations among the subgraphs. Shin et al [19] explored the relations between core (truss) degeneracy and the number of triangles. Our research directly addresses the relationship between the core and truss decompositions based on local perspectives. We also propose the CORE-TRUSSDD algorithm to detect the outlier nodes. Note that, in a prior work we also explored the relationship between clique counts and the core and truss degeneracy [20].

VIII. CONCLUSION

In this paper, we analyzed the interplay between core and truss decompositions in real-world networks. We introduced VI and EI plots to analyze the network structure by using the core and truss decompositions. The VI plot investigates the core-truss interplay from a vertex perspective, by examining the spectrum of edges around vertices with particular core numbers. The EI plot explores the interplay from an edge perspective by checking the truss number of an edge and the core numbers of the two endpoints. We applied our analysis on real-world networks from various domains, and then validate our findings by evaluating the random graphs generated by the BTER model. The VI and EI plots reveal consistent pattern in social networks and autonomous systems, and identify some interesting behaviors in collaboration networks, citation networks, and web networks.

Inspired by our observation in VI and EI plots, we proposed the CORE-TRUSSDD algorithm to identify anomalies in the networks by utilizing the core-truss interplay. We analyzed the characteristics of the outliers identified by our algorithm, and the results support our findings in the VI plots. We believe our study would be handy for domain practitioners to analyze the dense subgraph structure of networks, and provide important insights for anomaly detection.

Acknowledgments. This research was supported by NSF-1910063 award, JP Morgan Chase and Company Faculty Research Award, and Center for Computational Research at the University at Buffalo [21].

REFERENCES

- [1] X. Du, R. Jin, L. Ding, V. E. Lee, and J. H. T. Jr., "Migration motif: a spatial - temporal pattern mining approach for financial markets." in *KDD*, 2009, pp. 1135–1144.
- [2] D. Gibson, R. Kumar, and A. Tomkins, "Discovering large dense subgraphs in massive graphs," in *VLDB*, 2005, pp. 721–732.
- [3] E. Fratkin, B. T. Naughton, D. L. Brutlag, and S. Batzoglou, "Motifcut: regulatory motifs finding with maximum density subgraphs." in *ISMB*, 2006, pp. 156–157.
- [4] S. B. Seidman, "Network structure and minimum degree," *Social Networks*, vol. 5, no. 3, pp. 269–287, 1983.
- [5] K. Saito and T. Yamada, "Extracting communities from complex networks by the k-dense method," in *ICDMW*, 2006.
- [6] A. E. Sariyüce, C. Seshadhri, A. Pinar, and Ü. V. Çatalyürek, "Finding the hierarchy of dense subgraphs using nucleus decompositions," in *WWW*, 2015, pp. 927–937.
- [7] A. E. Sariyüce, C. Seshadhri, A. Pinar, and Ü. V. Çatalyürek, "Nucleus decompositions for identifying hierarchy of dense subgraphs," *ACM Transactions on the Web (TWEB)*, vol. 11, no. 3, p. 16, 2017.
- [8] V. Batagelj and M. Zaversnik, "An o(m) algorithm for cores decomposition of networks," Arxiv, Tech. Rep. cs/0310049, 2003.
- [9] J. Cohen, "Trusses: Cohesive subgraphs for social network analysis," vol. National Security Agency Technical Report, 2008.
- [10] C. Seshadhri, T. G. Kolda, and A. Pinar, "Community structure and scale-free collections of erdos-renyi graphs," *Phys. Rev. E*, vol. 85, p. 056109, May 2012.
- [11] J. Leskovec and A. Krevl, "SNAP Datasets," Jun. 2014.
- [12] M. Ley, "The dblp computer science bibliography: Evolution, research issues, perspectives," in *International symposium on string processing and information retrieval*. Springer, 2002, pp. 1–10.
- [13] J. Kunegis, "Konec: the koblenz network collection," in *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 2013, pp. 1343–1350.
- [14] P. Liu and A. E. Sariyüce, "Characterizing and utilizing the interplay between core and truss decompositions," <https://arxiv.org/abs/2011.00749>, 2020.
- [15] S. P. Borgatti and M. G. Everett, "Models of core/periphery structures," *Social Networks*, vol. 21, no. 4, pp. 375 – 395, 2000.
- [16] R. S. Burt, *Structural holes: The social structure of competition*. Harvard university press, 2009.
- [17] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.
- [18] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [19] K. Shin, T. Eliassi-Rad, and C. Faloutsos, "Patterns and anomalies in k-cores of real-world graphs with applications," *Knowledge and Information Systems*, vol. 54, no. 3, pp. 677–710, 2018.
- [20] P. Liu and A. E. Sariyüce, "Analysis of core and truss decompositions on real-world networks," 2019.
- [21] "Center for Computational Research, University at Buffalo," 2020, <http://hdl.handle.net/10477/79221>.