Weakening the Inner Strength: Spotting Core Collusive Users in YouTube Blackmarket Network

Hridoy Sankar Dutta*, **Nirav Diwan*** and **Tanmoy Chakraborty**
IIIT Delhi, India
hridoyd@iiitd.ac.in, nirav17072@iiitd.ac.in, tanmoy@iiitd.ac.in

Abstract

Social reputation (e.g., likes, comments, shares, etc.) on YouTube is the primary tenet to popularize channels/videos. However, the organic way to improve social reputation is tedious, which often provokes content creators to seek services of online blackmarkets for rapidly inflating content reputation. Such blackmarkets act underneath a thriving collusive ecosystem comprising *core users* and *compromised accounts* (together known as *collusive users*). Core users form the backbone of blackmarkets; thus, spotting and suspending them may help in destabilizing the entire collusive network. Although a few studies focused on collusive user detection on Twitter, Facebook, and YouTube, none of them differentiate between core users and compromised accounts.

We are the first to present a rigorous analysis of core users in YouTube blackmarkets. To this end, we collect a new dataset of collusive YouTube users. We study the *core-periphery structure* of the underlying *collusive commenting network* (CCN). We examine the topology of CCN to explore the behavioral dynamics of core and compromised users. We then introduce KORSE, a novel graph-based method to automatically detect core users based *only* on the topological structure of CCN. KORSE performs a weighted k-core decomposition using our proposed metric, called *Weighted Internal Core Collusive Index* (WICCI). However, KORSE is infeasible to adopt in practice as it requires complete interactions among collusive users to construct CCN. We, therefore, propose NURSE, a deep fusion framework that *only leverages user timelines* (without considering the underlying CCN) to detect core blackmarket users. Experimental results show that NURSE is quite close to KORSE in detecting core users and outperforms nine baselines.

Introduction

In recent years, YouTube has grown as a primary video-sharing platform, where content creators create channels and upload videos. The videos are then recommended to the content consumers based on several factors, one of which is the online *social reputation* of the creators and their content. Social reputation is usually quantified by the endorsement of the viewers in terms of likes, (positive) comments, shares, etc. However, an organic way of gaining reputation is a time consuming process, and often depends on several other factors such as the quality and relevance of the video, initial viewers and their underlying connections. Unfortunately, there exist a handful of online reputation manipulation services (*aka* blackmarkets) which help content creators rapidly inflate their reputations in an artificial way (Shah et al. 2017). Such services are built on a large thriving ecosystem of collusive network. The underlying network comprising *core users* – fake accounts or sockpuppets (Bu, Xia, and Wang 2013), which are fully controlled by the blackmarkets (puppet masters), and *compromised accounts* which are temporarily hired to support the core users – these two types of users are together called as *collusive users*. Core users are the spine of any collusive blackmarket; they monitor and intelligently control the entire fraudulent activities in such a way that none of their hired compromised accounts are suspended. Therefore, detecting and removing core blackmarket users from YouTube is of utmost importance to decentralize the collusive network and keep the YouTube ecosystem healthy and trustworthy. In this study, we deal with *freemium blackmarkets* (Shah et al. 2017) which invite customers to opt for the service for free, in lieu of surrendering their accounts temporarily for blackmarket activities. In doing so, customers gain virtual credit and use it to grow their content's reputation.

State-of-the-art and Motivation. Several efforts have been made to detect *fake activities* in different online social networks (Cresci et al. 2015; Castellini, Poggioni, and Sorbi 2017). However, as suggested by Dutta et al. (2021), *collusive activities* are very different from usual fake activities. A few studies attempted to explore the dynamics of blackmarkets, mostly for Twitter (Castellini, Poggioni, and Sorbi 2017; Dutta and Chakraborty 2020) and Facebook (Farooqi et al. 2017). On YouTube, there exists only one method, named CollATe to detect collusive users (Dutta et al. 2021). However, to our knowledge, none of these methods attempted to further divide collusive users into core and compromised accounts.

One may argue that once a collusive account (be it core or compromised) is detected, it should be banned. Then why do we need to explicitly identify core and compromised accounts, while both of them deserve punishment? We argue that the role of a core user is different from a compromised account in the collusive ecosystem; therefore, the extent of

*Both authors contributed equally to the paper.

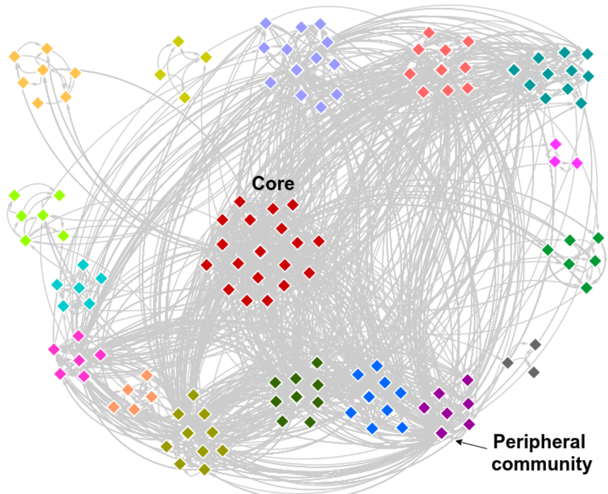


Figure 1: Visualization of the collusive commenting network (`CCN`). Unlike conventional core-periphery structure where peripheral nodes are sparsely connected internally, `CCN` constitutes dense peripheral communities sparsely connected with the core, indicating the growth of the network up to a certain point where it may not require core users to support compromised users for self-sustainability.

Notation	Denotation
$G(N, E)$	Collusive commenting network
V	Set of sets where $\{v_i\} \in \boldsymbol{V}$ indicates the set of videos created and posted by user n_i
$v_{i,j}$	j^{th} video in the video set $\boldsymbol{v_i}$
$comments(n, c)$	No. of comments posted by user n on video c
w_{ij}	Weight of the edge connecting nodes n_i and n_j
wc	weighted coreness score
$core_{th}$	Coreness threshold
G_C	Core subgraph
G_P	Induced subgraph of the peripheral nodes
G_P^L	Largest connected component in G_P
$WCS_{Core,C}$	Weighted cut set between the core and a peripheral community C

Table 1: Important notations and denotations.

punishment may differ. Compromised users are more interested in self-promotion; they join blackmarkets temporarily; they gain appraisals for their online content both organically (genuine interest by other users) and inorganically (through blackmarket services). However, core users, being the backbone of the blackmarkets, always intend to grow and popularize their business. They are permanent members of the blackmarkets; they provoke other users to join the services; and they generally initiate the artificial inflation of the reputation of online content. Therefore, they are more harmful to pollute the online ecosystem. Due to such contrasting behavior of core and compromised users, one may consider that core users should be punished differently than compromised users. For instance, a complete ban of core users would limit the growth of the collusive blackmarkets. However, for compromised users, it may be wise to just warn them and restrict their social network activities for a limited time, instead of a complete ban. The authorities of a social media platform may design suitable policies to handle these two cases.

To our knowledge, **ours is the first attempt to identify and explore the dynamics of *core blackmarket users*. It is also the second attempt after** `CollATe` **(Dutta et al. 2021) to explore *YouTube blackmarkets*.**

Present Work: `KORSE`. In this paper, we investigate the dynamics of core users in YouLikeHits, one of the popular YouTube blackmarket services. We start by collecting a novel dataset from YouLikeHits and YouTube, consisting of collusive users, the videos they promote through blackmarkets, and their comments on YouTube videos. In this study, we deal with only one type of appraisals i.e., *collusive comment* on YouTube videos. We then construct a collusive commenting network (`CCN`) based on the co-commenting activities among collusive users. We leverage the topological structure of `CCN` to detect core users using our proposed method, `KORSE` which utilizes k-core decomposition particularly designed based on our proposed metric, *Weighted Internal Core Collusive Index* (`WICCI`).

Present Work: Core-periphery Structure. An exhaustive analysis on the interactions of core and peripheral nodes reveals a counter-intuitive core-periphery structure of `CCN` – unlike a conventional network where peripheral nodes are sparsely connected, and get disconnected upon removal of the core, `CCN` constitutes peripheral nodes which form several small and dense communities around the core (c.f. Fig. 1). We further observe that there exists a strong positive correlation between the internal interactions within peripheral communities and the interactions between the core and the peripheral communities. This gives us the evidence that in peripheral communities, compromised users who comment heavily on videos that are co-commented by core users, tend to contribute more to the collusive market. We also present a case study to highlight the major differences between core and compromised users based on their user timelines: (i) Core users, although act as heavy contributors of the blackmarket services, are not the top beneficiaries of the collusive market. (ii) Core users indulge in less self-promotion of videos. (iii) Core users are less active participants of the collusive market than compromised users; they initiate the fraudulent activities and let the compromised users finish the remaining job.

Present Work: `NURSE`. Although `KORSE` is highly accurate in detecting core users, it is practically infeasible to deploy as it requires the complete snaphot of the collusive market on a streaming basis and is also required to be re-run on the introduction of each new user. Therefore, we consider core users detected by `KORSE` as the ground-truth[1] and de-

[1]Collecting the ground-truth for fake/genuine entity detection is challenging, which usually requires annotations from annotators with domain expertise (Shu, Wang, and Liu 2018). However, obtaining the ground-truth data of core blackmarket users is almost impossible. We do not know any legal way to find "core" blackmarket users. Therefore, we consider `KORSE` as an oracle, which cannot be used in practice but can be used to create the ground-truth.

velop NURSE, a deep fusion framework that *only* considers user timeline (without the underlying CCN) and video submission information to detect core blackmarket users. Experiments on our curated dataset show that NURSE is quite close to KORSE with 0.879 F1-Score and 0.928 AUC, outperforming nine baselines.

Contributions: In short, our contributions are four-fold:

- **Novel problem:** We are the first to address the problem of *core blackmarket user* detection.
- **Unique dataset:** Our curated dataset is the first dataset, comprising *core and compromised collusive YouTube users*.
- **Novel methods:** Our proposed methods, KORSE and NURSE, are the first in detecting core blackmarket users.
- **Non-intuitive findings:** Empirical analysis of the dynamics of core and compromised users reveals several non-trivial characteristics of blackmarket services.

Reproducibility. Our full code and dataset are available here - https://github.com/LCS2-IIITD/ICWSM-2022-Core-Collusive-Youtube-BlackMarket

	Batagelj and Zaversnik (2003)	Shin, Eliassi-Rad, and Faloutsos (2016)	Cheng et al. (2011)	Rombach et al. (2014)	Zhang et al. (2017)	Dutta et al. (2021)	KORSE	NURSE
Detect collusive users						✓	✓	✓
Detect core blackmarket users							✓	✓
Graph-based approach	✓	✓	✓	✓	✓		✓	
Deal with weighted graph	✓			✓			✓	
Consider profile information						✓		✓
Consider content information						✓		✓

Table 2: Qualitative comparison of KORSE and NURSE with similar approaches.

Related Work

We summarize related studies by dividing them in two subsections: (i) blackmarkets and collusion, and (i) network core detection.

Blackmarkets and Collusion: Recently, the activities of blackmarket services have garnered significant attention among the researchers due to the way they provide artificial appraisals to online media content. Shah et al. (2017) provided a broad overview of the working of blackmarkets. Dutta and Chakraborty (2020); Dutta et al. (2018, 2020) attempted to detect collusive retweeters on Twitter. The authors also mentioned how collusive users are asynchronous in nature as compared to normal retweet fraudsters. Dutta and Chakraborty (2020) further studied the working of premium and freemium blackmarket services in providing collusive appraisals on Twitter. Arora et al. (2020) further investigated the blackmarket customers engaged in collusive retweeting activities using a multiview learning based approach. Chetan et al. (2019) proposed CoReRank, an unsupervised method to detect collusive retweeters and suspicious tweets on Twitter. Farooqi et al. (2017) showed how collusion networks collect OAuth access tokens from colluding members and abuse them to provide fake likes or comments to their members. Dhawan et al. (2019) proposed DeFrauder, an unsupervised framework to detect collusive behavior of online fraud groups in customer reviews. We encourage the readers to go through Dutta and Chakraborty (2020) for a comprehensive survey on blackmarket-based collusive activities in online media platforms. Dutta et al. (2021) is the closest to the current research, which detects collusive blackmarket users on YouTube. However, it does not focus on detecting *core blackmarket users*.

Network Core Detection: Due to the abundance of literature on network core detection, we restrict our discussion to some selected works that we deem as pertinent to our study. k-core decomposition (Batagelj and Zaversnik 2003) is considered to be the *de facto* to detect core nodes. It is based on the recursive removal of vertices that have degree less than k in the input network. Rombach et al. (2014) proposed an algorithm to detect core-periphery structure in networks. The goal of this algorithm is to identify densely connected core nodes and sparsely connected peripheral nodes. Cucuringu et al. (2016) detected core and periphery using spectral methods and geodesic paths. Kojaku and Masuda (2017) discovered multiple non-overlapping groups of core-periphery structure by maximizing a novel quality function which compares the number of edges of different types in a network. (Xiang et al. 2018) detected multiple core-periphery structures and communities based on network density. The authors also proposed an improved version of their model to detect active and overlapping nodes. Zhang et al. (2017) studied the problem of collapsed k-core to identify a set of vertices whose removal can lead to the smallest k-core in the network. Shin, Eliassi-Rad, and Faloutsos (2016) showed empirical patterns in real-world graphs related to k-cores. Recently, it has been observed that the subgraphs of the detected core users are used for several graph-related tasks, such as community detection (Peng, Kolda, and Pinar 2014), dense-subgraph detection (Hooi et al. 2020) etc. We encourage the readers to go through Malliaros, Papadopoulos, and Vazirgiannis (2016) for a comprehensive survey on network core detection.

Differences with Existing Studies: Table 2 compares our methods (KORSE and NURSE) with a few relevant studies. In short, our methods are different from others in five aspects – (i) we are the first to address **core blackmarket user detection** problem; (ii) we are the second after (Dutta

One can argue that the current way of creating the ground-truth may be unconvincing. However, we perform several case studies to provide strong empirical evidence which may validate our strategy of collecting the ground-truth. We do not know any other way of ground-truth creation for this problem unless blackmarkets themselves provide the same!

et al. 2021) to deal with **YouTube** collusive blackmarkets; (iii) We propose **both unsupervised** (KORSE) and **supervised** (NURSE) methods for core detection; (iv) our **dataset comprising core blackmarket users** is unique; and (v) we provide a **rigorous analysis** to explore the dynamic of core and compromised users.

Methodology

Dataset Description

In this work, we consider YouLikeHits[2], a freemium blackmarket service[3]. We designed web scrapers to extract the ids of YouTube videos submitted to blackmarket services for collusive comments. We used YouTube API[4] to extract the metadata details and comment history of these videos. We extracted $26,166$ YouTube videos which were submitted to YouLikeHits for collusive comments. These videos were uploaded to $11,000$ unique YouTube channels. To our knowledge, this is the first dataset of its kind. Note that the entire data collection process was performed after taking proper Institutional Review Board (IRB) approval.

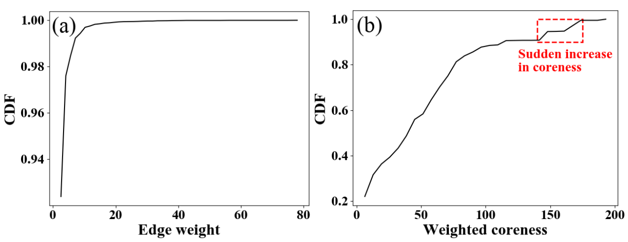


Figure 2: Cumulative distribution of (a) edge weights, and (b) weighted coreness scores of nodes in CCN. Contrary to the general observation that coreness score follows power law, we observe that there are relatively large number of nodes having high weighted coreness.

Preliminaries and Graph Construction

Here we present some important concepts used throughout the paper. Table 1 summarises important notations.

[**Collusive Users and Videos**] We define *collusive users* as those who are involved in the blackmarket activities. There are two types of collusive users – core users and compromised users. We call the videos submitted to freemium blackmarkets as collusive videos.

[**Core Users**] A limited set of online accounts are fully controlled by the blackmarket authorities. These accounts can be bots (fully automated), sockpuppets (controlled by puppet masters) (Bu, Xia, and Wang 2013) or fake accounts. However, they are used only to benefit blackmarkets. We call these users core blackmarket users.

Property	Value
# nodes	$1,603$
# edges	$51,424$
Avg./max/min edge weight	1.392 / 78 / 1
Avg./max/min weighted degree of nodes	89.367 / 1638 / 1
Unweighted edge density	0.040
Unweighted clustering coefficient	0.737
Network diameter	8

Table 3: Topological properties of CCN.

[**Compromised Users**] These are YouTube content creators who submit their content to the freemium blackmarkets in order to receive artificial comments within a short duration. Being freemium customers, their accounts are compromised for a limited time to perform illegal activities by commenting on videos of other blackmarket customers.

[**Collusive Commenting Network (CCN)**] A CCN is an undirected and weighted network $G(N, E)$, where each node $n \in N$ represents a collusive user, and two nodes n_i and n_j are connected by an edge $e_{ij} = \langle n_i, n_j \rangle$ if the corresponding users co-commented on the same videos. The weight w_{ij} of the edge e_{ij} is calculated as per Eq. 1.

Let us denote a set of sets, $V = \{\{\boldsymbol{v_1}\}, \{\boldsymbol{v_2}\}, \{\boldsymbol{v_3}\}, \dots\}$, where $\{\boldsymbol{v_i}\}$ indicates the set of videos posted by collusive user n_i. $\{\boldsymbol{v_{i,j}}\}$ indicates the j^{th} video in the set $\boldsymbol{v_i}$.

[**Inter-user Comment Count**] The number of comments posted by the collusive user n on video c is denoted by $comments(n, c)$. We define *Inter-user comment count* (IUCC) for a video c and a pair of users n_i and n_j as the minimum of the number of comments by n_i and n_j on c.

$$IUCC(n_1, n_2, c) = min\big(comments(n_1, c), comments(n_2, c)\big) \tag{1}$$

[**Edge weight**] We measure the *edge weight* between two nodes (collusive users) n_i and n_j as follows:

$$w_{ij} = \sum_{\substack{p=1\\ p\neq i,j}}^{|V|} \sum_{q=1}^{|v_p|} IUCC(n_i, n_j, v_{p,q}) \tag{2}$$

The edge weight w_{ij} indicates the aggregated IUCC across all the videos co-commented by n_i and n_j, excluding their own videos. We exclude the videos created by n_i and n_j since the comments on these videos can be easily manipulated (added or deleted) by the owners themselves. Table 3 summarises the properties of CCN. Fig. 2(a) shows the cumulative distribution of w_{ij}.

Weighted k-core Decomposition

Given a graph $G(N, E)$, the weighted k-core detection problem aims to find *k-core* (or core of order k), the maximal induced subgraph denoted by $G_k(N_k, E_k)$ such that $G_k \subseteq G$ and $\forall n \in N_k : deg(n) \geq k$. The following two methods are often used to solve this problem: *k-core decomposition* (Rombach et al. 2014) and *core-periphery algorithm* (Della Rossa, Dercole, and Piccardi 2013). In our case, we

[2] https://www.youlikehits.com/

[3] Freemium blackmarkets offer customers to enjoy their services for free with the condition that the customers will temporarily act on behalf of the blackmarkets. Upon signing up, the social media accounts of customers are compromised for a limited time for blackmarket activities, which in turn help them gain virtual credits

[4] https://developers.google.com/youtube/v3

choose *k-core decomposition*[5]. In (weighted) k-core decomposition, to detect core users, we repeatedly delete nodes with (weighted) degree[6] less than k until no such node is left (this is also known as "shaving" method (Shin, Eliassi-Rad, and Faloutsos 2016)). The reasons behind choosing k-core decomposition are as follows: (i) It has been empirically shown to be successful in modeling user engagement (Zhang et al. 2016, 2017); (ii) Unlike k-core, core-periphery algorithm fits more closely with networks where the nodes are not closely connected to each other (Borgatti and Everett 2000). However, in blackmarket services, the sole purpose of collusive users to join the services is to gain credits (by providing collusive appraisals to the content of other users) which can be used by them to artificially inflate their social growth. This strengthens the connectivity among the collusive users. The reason behind expecting high interactions among users stems from the fact highlighted in (Dutta et al. 2021) that different collusive users retweet the same tweets on the collusive market regardless of the topic of the tweets. We expect a similar behavior in case of YouTube comments, i.e., different collusive users tend to comment on the same videos in order to earn credits. In our dataset, a collusive video has an average of 3 comments by collusive users. This would create more relations (edges) between nodes in CCN.

WICCI: Expected Behavior of Core Users

We frame the core detection problem in CCN as the weighted k-core decomposition problem in CCN. k-core decomposition assigns a *coreness value* to each vertex. In our case, the coreness value ranges from 1 to 193, with an average value of 48.7. We obtain an ordered list of vertices sorted in decreasing order of the coreness value. Typically, the node assigned with the highest coreness value is said to be the "most influential node" in the graph. The subgraph formed with such highly influential vertices is known as *degeneracy-core* or k_{max}-core. On running the weighted k-core decomposition on CCN, we obtain a *degeneracy-core* consisting of 8 users. We expect the distribution of nodes to continually decrease with increasing coreness, as observed in typical core-periphery structures. However, we observe that the fraction of nodes with a high weighted coreness is unusually high (12.1% users with ≥ 100 coreness score as shown in Fig. 2(b)). This indicates the presence of *a larger set of core users*.

Therefore, in CCN, to define the partition of core and compromised users, we propose a metric, called *Weighted Internal Core Collusive Index* (WICCI) which is motivated by Rombach et al. (2014). WICCI is used to partition the list of decreasing weighted coreness values by a "coreness threshold". The nodes whose coreness is above the threshold are eligible to be the core nodes, while the remaining nodes are considered as compromised users. To define WICCI, we consider two important properties of core users as follows:

1. **Density:** A core component of a network should be densely connected (Rombach et al. 2014; Borgatti and Everett 2000). We attempt to understand the implications of a dense core in CCN, by considering the flip-side first – *a sparse core*. A sparse core in CCN would have less number of edges connecting vertices internally. In the current scenario, it implies that different users have commented upon different sets of videos. However, the existence of such an entity would mean that there is no cohesion or strategy in the way core users operate. They may be commenting randomly on different videos. The existence of a dense core, however, would imply that different users are commenting on a same set of (collusive) videos, indicating some cohesion or strategy. Note that when we increase the coreness threshold, the subgraph of the core formed has an increasing density (and a decreasing size).

$$\texttt{WICCI} \propto density^{\beta} \tag{3}$$

where β is the density coefficient. We utilize β to vary the proportionality of WICCI with density.

2. **Fraction of weighted size of core:** There is a major flaw in considering only density to define a core. Density does not take into account the edge weight i.e., the volume with which the two users have commented together on same videos. We intuitively expect that inside a core, a high fraction of the commenting activities take place. We define W_G as the weighted size (sum of the weights of edges) of CCN and W_C as the weighted size of the core subgraph G_C. Correspondingly,

$$\texttt{WICCI} \propto \frac{W_C}{W_G} \tag{4}$$

Combining (3) and (4), we get

$$\texttt{WICCI} = k \times \frac{W_C}{W_G} \times density^{\beta} \tag{5}$$

where k is the constant of proportionality. We assume it to be 1.

KORSE: A Graph-based Method for Core Detection

By considering the above properties of collusive entities, we design KORSE (**K**-core decomposition for c**OR**e collu**S**ive us**E**rs detection), a modified version of (weighted) k-core decomposition that is designed for detecting core users in blackmarket services based only on the topological structure of CCN. It takes CCN as input and detects core blackmarket users (core subgraph G_C). KORSE is implemented by decreasing the coreness threshold and consequently making larger subgraphs of the core. The subgraph with the largest WICCI is our final core.

Algorithm 1 presents the pseudo-code of KORSE. Firstly, we apply weighted k-core decomposition which gives the weighted coreness score $wc(n)$ for each vertex $n \in N$. The vertices are then sorted in decreasing order of wc and pushed into a stack $\mathcal{S}$. The top of the stack is the node with the maximum weighted coreness. Next, we create a running set ($core_n$) of core nodes initially with no node. The running coreness threshold $core_{th}$ is set to the maximum value

[5] We use the weighted version of *k-core decomposition* to incorporate the edge weights (see Eq. 1 for more details).

[6] The weighted degree of a node is the sum over the edge weights of the connected edges.

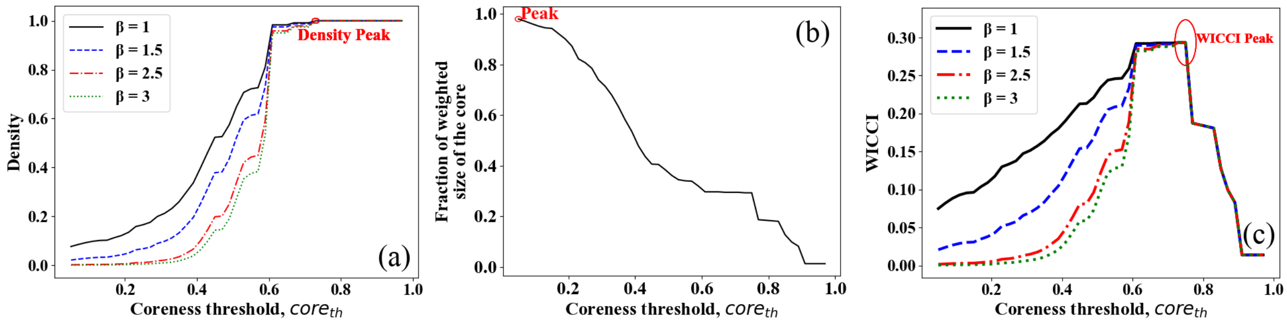


Figure 3: Variation of (a) density, (b) fraction of weighted size of core, and (c) `WICCI` with varying $core_{th}$. (a) Initially, the density dominates over fraction of weighted size of the core and hence `WICCI` increases rapidly in (c). (b) In the later stages, the inverse happens – fraction of weighted size of core dominates which results in `WICCI` declining steeply in (c). The `WICCI` peak of 0.294 is observed at $core_{th} = 0.73$ in (c). All nodes with a weighted coreness above $core_{th}$ are part of the core. Note that despite varying the density of core w.r.t `WICCI` by changing the density coefficient (β), we observe a similar `WICCI` peak in all cases.

of weighted coreness wc_{max}. Next, $core_{th}$ is iteratively decreased, and the set of core nodes is updated by adding all nodes n which have $wc(n)$ greater than $core_{th}$. Next, G_{cr}, an induced subgraph is created only by the core nodes. Further, `WICCI` of G_{cr} is calculated. The induced subgraph with the maximum `WICCI` ($wicci_{max}$) is the core of the graph, and the corresponding $core_{th}$ is the coreness threshold.

On applying `KORSE` on `CCN`, we obtain an ideal coreness threshold of 0.73 on a max-normalized scale, with a peak `WICCI` value of 0.294 (c.f. Fig. 3) for different values of the density coefficient β. We explore the variation of `WICCI` with $core_{th}$:

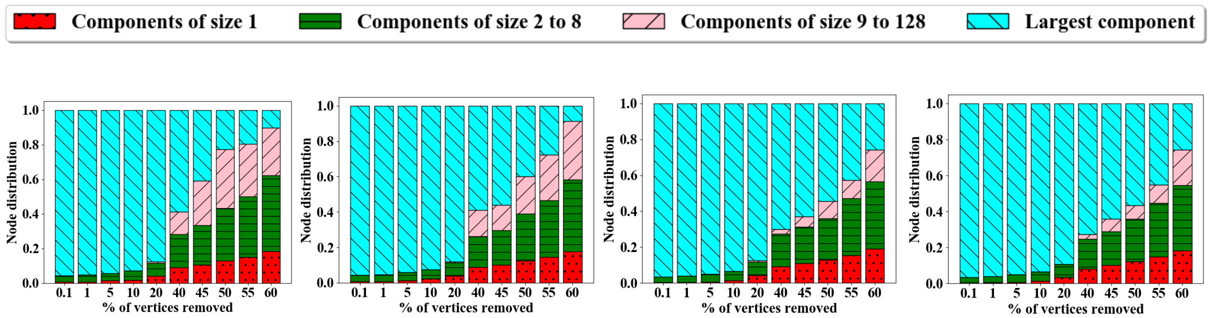
1. Initially, as $core_{th}$ increases ($0.1 - 0.5$), users of low wc (which contribute less to the overall collusive activity of the network) are removed from the core subgraph, leading to rapid increase in density of the core subgraph and a relatively smaller decrease in the fraction of weighted size of the core. Initially, density dominates the fraction of weighted size of the core and hence `WICCI` increases (c.f. Fig. 3(a)).
2. Towards the higher values of $core_{th}$ (> 0.8), density obtains its maximum value of 1. However, the fraction of weighted size of the core decreases rapidly due to the continued exclusion of more nodes with relatively higher wc. As $core_{th}$ increases further, the fraction of weighted size of the core dominates density towards the latter values of $core_{th}$, and hence `WICCI` decreases (c.f. Fig. 3(b)).
3. In the mid-range values ($0.6 - 0.7$) of $core_{th}$, the peak of `WICCI` is observed. The corresponding core formed by the nodes (with wc higher than $core_{th}$) leverages both the density and the fraction of weighted size of `CCN` (c.f. Fig. 3(c)).

The core obtained on applying `KORSE` consists of 148 nodes and (surprisingly) *is a complete graph*. Nearly 30% of the entire collusive commenting activities of the network happens among 10% of the core nodes. The periphery consists of $1,455$ nodes and has an edge density of 0.0355. Nearly 60% of the commenting activities take place among the peripheral nodes despite 90% of the users belonging to it. The rest 10% activities are captured between the core and the peripheral nodes (cross-edges between core and periphery). We now investigate the connectivity of the core in our proposed CCN network.

Impact of Core on `CCN`

To closely explore the connectivity of the core in the network, we analyse the effect after removing the core from `CCN`. Mislove et al. (2007) reported that in a conventional social network, the removal of core breaks the graph into small disconnected components. However, in our case we notice that the graph does not break into smaller components even after removing a large fraction of core nodes (c.f. Fig. 4). The possible reasons for such a behavior are as follows:

1. **Estimated core may be incorrect:** One may argue that our metric `WICCI` to estimate the core may be flawed. It may be possible that the core is larger than what we estimate. To verify this, we start by removing the vertices from `CCN` in the decreasing order of the (i) weighted degree (c.f. Fig. 4(a)), and (ii) weighted coreness wc (c.f. Fig. 4(c)). We observe that the point where the size of the largest connected component decreases and the number of small disconnected components increases drastically, should be the appropriate value of $core_{th}$. However, we notice that such a point arises only after removing 50% and 60% of nodes based on weighted degree and weighted coreness of vertices from `CCN`, respectively. This would suggest that at least 50% of the vertices belong to the core. However, the density of the core reduces significantly (c.f. Fig. 5). This violates one of the fundamental properties of a core that it should be incredibly dense. Mislove et al. (2007) observed near-complete degradation of the largest connected component after only removing 10% of the nodes based on de-



(a) Weighted degree (b) Unweighted degree (c) Weighted coreness (d) Unweighted coreness

Figure 4: The distribution of nodes in components of sizes present in `CCN` after removing nodes in the decreasing order of (a) weighted degree, (b) unweighted degree, (c) weighted coreness, and (d) unweighted coreness. The network visibly disintegrates into smaller components when at least (a) 50% (b) 55% (c) 60%, and (d) 60% are removed from the network. Despite a large removal of nodes, the remaining network has a high connectivity.

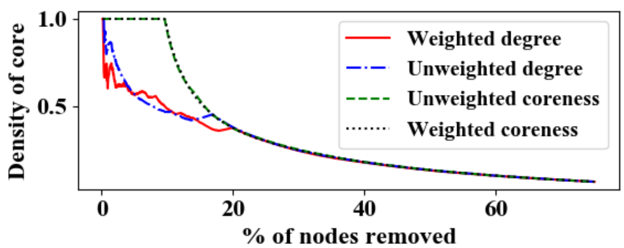


Figure 5: Change in the density of core with the number of nodes removed.

gree. Therefore, the observed pattern is not the artifact of our proposed metric `WICCI`, but a result of the high connectivity even among users of low coreness.

2. **Weighted k-core decomposition may be incorrect:** One may argue that we should consider the traditional *unweighted* k-core decomposition (Mislove et al. 2007), instead of considering the weighted edges. We perform similar experiments by removing vertices in the order of the (i) unweighted degree (c.f. Fig. 4(b)) (as suggested in Mislove et al. (2007)) and (ii) unweighted coreness (c.f. Fig. 4(d)). We observe similar results in both the cases where the network breaks into many small disconnected components upon removing at least 55% of the nodes. This would again make the core incredibly sparse (c.f. Fig. 5). Therefore, applying weighted k-core decomposition is not a reason for the late disintegration of the graph into smaller components.

Possible explanation: connected periphery. We examine G_P, the induced subgraph of the peripheral nodes independently with specific focus on its largest connected component G_P^L.

- G_P^L and G_P have $1,376$ and $1,455$ nodes, respectively.
- G_P^L and G_P have edge density of 0.03674 and 0.0355, respectively.
- G_P^L has an average path length of 2.6355.
- Lastly, as stated earlier, when we progressively remove the core from `CCN`, the periphery largely remains intact.

This indicates that there is a significant connectivity among nodes in the periphery. This does not fall within the conventional structure of the periphery which is generally described as small disconnected components. Instead, we visualize the periphery in G_P^L as smaller and relatively dense communities (c.f. Fig. 1). One possible reason for a connected periphery may be that the graph has organically grown to a stage where despite the detection of the core users, the blackmarket service is in a self-sustainable stage and is no longer driven by the core users alone. A solution would be to detect the core users at an early stage to halt the growth of the market. To identify the network at its infancy, one would have to create multiple snapshots of the blackmarket services over a period of time, which is a computationally expensive task. We now examine the relation between the core and peripheral communities present in the proposed network.

Interplay Between Core and Peripheral Communities

Here, we study the interactions between the core and periphery, and highlight critical observations. We start by dividing the videos V into three categories:

1. **Core-core videos** are the set of videos commented exclusively by core users.
2. **Core-periphery videos** are the set of videos commented by both core and peripheral users.
3. **Periphery-periphery videos** are the set of videos commented exclusively by peripheral users.

Here, (1) and (2) are responsible for the formation of edges within core; (2) and (3) are responsible for the formation of edges within periphery; (2) alone is responsible for the formation of edges between core and periphery.

Next, we define the community structure in `CCN`. A "good" community in `CCN` is the one in which the users of the community have co-commented heavily on a set of videos. Due to the high connectivity observed in the periphery (mentioned in the earlier section), we speculate that the

Algorithm 1: KORSE algorithm

Input: CCN $G(N, E)$
Output: G_c: Subgraph containing core nodes
Initialize $wicci_{max} \leftarrow 0$;
▷ *Running the weighted k-core decomposition on $G(N, E)$.*
wc = List of weighted coreness scores for nodes in $G(N, E)$.
▷ *Sort N by wc and push into stack S*
$\mathcal{S}$ = Stack of nodes in G in descending order of weighted coreness, wc.
▷ *Running set of core nodes*
$core_n \leftarrow [\,]$
▷ *Set coreness threshold as the max weighted coreness*
$core_{th} \leftarrow max(wc)$
while $core_{th} > 0$ **do**
  ▷ *Get node with maximum coreness*
  $n = \mathcal{S}.pop()$
  while $wc(n) >= core_{th}$ **do**
    ▷ *Add n to $core_n$*
    $core_n.add(n)$
    $n = \mathcal{S}.pop()$
  end
  ▷ *As $wc(n) < core_{th}$, we push n back to $\mathcal{S}$*
  $\mathcal{S}.push(n)$
  ▷ *Make induced subgraph of core using current $core_v$*
  $G_{cr} \leftarrow$ InducedSubgraph(G, $core_n$)
  ▷ *Compute WICCI for the current core G_{cr}*
  wicci $\leftarrow$ WICCI(G_{cr},G)
  ▷ *Finding the G_{cr} with maximum WICCI*
  if $wicci > wicci_{max}$ **then**
    $wicci_{max} \leftarrow$ wicci
    $G_c = G_{cr}$
  end
  ▷ *Iteratively decrease the coreness threshold*
  $core_{th} \leftarrow core_{th} - 1$
end

periphery consists of several small communities. To check this, we run the weighted version of the Louvain community detection method (Blondel et al. 2008) for detecting peripheral communities C_P^L from G_P^L (the largest connected component in the induced subgraph in the periphery). The modularity of the community structure detected by Louvain is 0.397, and the number of large communities (with size > 40) is 9. It indicates that there exist large communities of collusive users that comment on the same set of videos. Next, we define the interaction within the peripheral community based on the amount of collusive commenting activities occurring inside the community. We categorize these interactions using (a) weighted size, and (b) average weighted degree of nodes in the peripheral community. We also quantify the interactions between core and each of the peripheral communities based on the amount of commenting activities

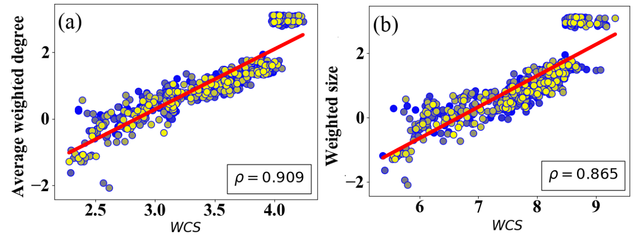


Figure 6: A strong positive correlation between weighted cut-set WCS and – (a) average weighted degree, and (b) weighted size of the peripheral communities. Different colors indicate communities obtained in different executions of Louvain method. The Pearson's ρ is also reported.

on the core-periphery videos.

[**Internal Interaction of Peripheral Community**] We define the internal interaction of a peripheral community as a measure of the collusive commenting activities within the community.

We further categorize the internal interaction using the following metrics:

1. ***Average weighted degree of nodes in the community***: It captures the average collusive commenting activities taking place within the community.
2. ***Weighted size of the community***: It is measured by the sum of weights of all the internal edges of a community, capturing the total intra-community collusive commenting activities.

[**Independent Interaction of Core and Peripheral Community**] We define the independent interactions of core and a peripheral community as a measure of the collusive commenting activities taking place between the core and the peripheral community. This indicates the participation of the peripheral users in commenting on core-periphery videos.

To capture independent interactions between core and peripheral community C, we utilize the **weighted cut-set** $WCS_{Core,C}$ as the sum of the weights of edges connecting the core and C. Since the size of the peripheral communities varies, we normalize $WCS_{Core,C}$ by only $|C|$.

$$WCS_{core,C} = \frac{\textit{Sum of weights of edges connecting core and C}}{|C|}$$

The following observations are drawn from the above (c.f. Fig 6):

1. There exists a positive correlation between the average weighted degree of a peripheral community and $WCS_{core,C}$ (c.f. Fig 6(a)).
2. There exists a positive correlation between the weighted size of a peripheral community and $WCS_{core,C}$ (c.f. Fig 6(b)).

From these observations, we conclude that there is a definite positive correlation between the internal interaction within the peripheral communities and that between the core and peripheral communities. Peripheral communities which actively participate in activities associated with the core (such as commenting on core-periphery videos), tend to contribute more to the collusive market. We now discuss in our detail our proposed deep fusion framework `NURSE` for the identification of core blackmarket users.

`NURSE`: A Deep Fusion Framework

Although the network topology based weighted k-core decomposition presented in `KORSE` is highly accurate to detect core blackmarket users, it may not be feasible to adopt in designing a real-world system because of the following reasons: (i) data arrives in streaming fashion, and the generation of `CCN` is not possible as the entire snapshot of the blackmarkets at a certain point is impossible to collect; (ii) `CCN` is often incomplete and highly sparse, and (iii) k-core decomposition is comparatively slow. **However, we consider `KORSE` as an oracle and the core and compromised users it has detected as the ground-truth to train and evaluate the following model.** To address the above issues and towards designing a real-world system, we propose `NURSE` (**NeU**ral framework for detecting co**R**e collu**S**ive us**E**rs), a neural fusion model to detect core blackmarket users in blackmarket services *based only on the user timeline and video sharing information* (without considering the underlying `CCN`).

`NURSE`: Model Components

`NURSE` comprises three components: *metadata feature extractor* **(MFE)**, *similarity feature extractor* **(SFE)**, and *textual feature extractor* **(TFE)**; the output of which are further concatenated to form the feature representation of a YouTube user. The combined representation is passed through to a *core detector* module which determines whether the user is a core or a compromised user. The architectural diagram of `NURSE` is shown in Fig. 7. Individual components of `NURSE` are elaborated below.

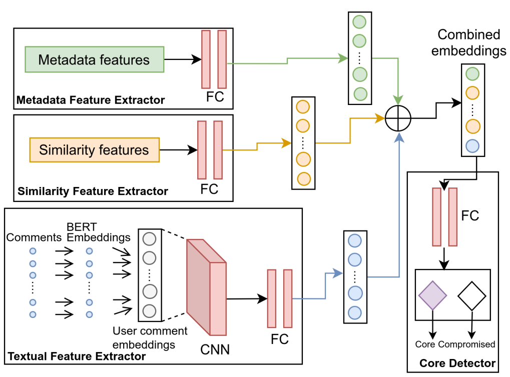


Figure 7: A schematic diagram of `NURSE`. The green colored network is the *metadata feature extractor* **(MFE)**, the orange colored network is the *similarity feature extractor* **(SFE)**, and the blue colored network is the *textual feature extractor* **(TFE)**. We concatenate the output of the feature extractors to form the feature representation of a YouTube user. The final representation is passed through to a *core detector* module to detect whether the given user is a core user or a compromised user.

Metadata Feature Extractor (MFE). We extract 26 metadata features based on the profile information, and videos uploaded by the users. These features are largely divided into four categories:

(a) **Self-comments** (MFE_{1-5}): These features are derived from the comments made by the users on their own videos. We observe that, on an average, compromised users tend to write more self-comments ($\times 1.778$) than the core users, indicating that *core users are less involved in self-promotion*. We take the maximum, minimum, total, average and variance of the comments across self-posted videos as five different features.

(b) **Number of videos uploaded** (MFE_6): It refers to the total number of videos uploaded by the user. On average, core users upload fewer videos, which is $\times 0.633$ less than that of compromised users. *A core user's efforts to benefit from the blackmarkets are lesser* (as they are created by the blackmarket services themselves) than the compromised users.

(c) **Duration of uploaded videos** (MFE_{7-11}): These features measure the duration of the videos uploaded by users. On average, a core user uploads significantly shorter videos, which is $\times 0.628$ less than that of compromised users. The possible reason could be that *core users are less interested in their own content*; rather their primary objective is to artificially inflate the popularity of other customers' videos. We take the maximum, minimum, total, average and variance of video duration per user as five different features.

(d) **Other features:** Apart from the above features, we also consider the following features related to the rating of the videos posted by a user (in each case, we take the maximum, minimum, total, average and variance as five different features) – the number of likes (MFE_{12-16}), the number of dislikes (MFE_{17-21}) and the number of views received (MFE_{22-26}).

Similarity Feature Extractor (SFE). Collusive users have been shown to post similar/duplicate comments regardless of the topic of the content (Dutta et al. 2021). We extract two sets of features based on the linguistic similarity of comments posted on the video and video metadata:

(a) **Comment-based features**: We capture similarity features based on the linguistic similarity of comments posted by users. For a user, let the set of her comments on her own videos and the set of comments on other videos be SC and OC, respectively. We first generate embedding of individual comments using pre-trained BERT (Devlin et al. 2018). We then measure the maximum, minimum, total, average and variance of similarities (cosine similarity) between comments in SC. Similarly, we obtain five similar features, each from the comments within OC and by comparing comments in SC and OC. This results in 15 features (SFE_{1-15}).

(b) **Video metadata based features**: In YouTube, a user can upload her own videos (SV) or act on videos posted by other users (OV). For each video, we combine the text of the video title, video description and video genre. We then generate the embedding of the combined text using BERT. Next, we extract the maximum, minimum, total, average and variance of similarities (cosine similarity) between video embeddings, each from the videos within SV and and videos across SV and OV. This results in 10 features denoted by SFE_{16-25}. We did not extract features from within OC because we observed that doing so heavily biased the model.

Textual Feature Extractor (TFE). We capture textual features from the content of the comments posted by a user. We generate embeddings for every comment using pre-trained BERT (Devlin et al. 2018). To get a representative embedding for a user, we average out the embeddings of all the comments posted by the user. As collusive users tend to post repetitive text in their comments (Dutta and Chakraborty 2020), we feed the resultant embedding into a CNN to capture this inter-dependency. In literature, CNNs have shown to perform well in capturing repetitive patterns in texts (Lettry et al. 2017).

Core Detector. The core detector module consists of a fully-connected layer (FC) with softmax to predict where a YouTube user is core or compromised, denoted by $G_c(., \theta_c)$, where θ_c represents the model parameters. For the prediction task, G_c generates the probability of a user u being the core user based on the combined representation $\vec{u}$.

$$P_\theta(u) = G_c(\vec{u}; \theta_c) \tag{6}$$

We use the cross-entropy loss (L_d) for our model:

$$L_d(\theta) = y \log \big(P_\theta(u)\big) + (1-y) \log \big(1 - P_\theta(u)\big) \tag{7}$$

NURSE: Model Specifications

NURSE executes three parallel operations - **(1) TFE:** The 1×784 textual vector is fed to a CNN (number of channels = 32, filter size = 2, no padding). Next, the resultant vector is passed to a max-pooling layer and then to a FC layer of size 64. The final output from this operation is a 1×64 vector. **(2) SFE:** The 1×25 similarity vector is fed to a FC Layer of size 32. A dropout of 0.3 is applied on the FC layer. The final output from this operation is a 1×32 vector. **(3) MFE:** The 1×26 metadata vector is passed to a FC layer of size 16. A dropout of 0.25 is applied on the FC layer. The final output from this operation is a 1×16 vector.

The combined representation is a 1×112 vector. This is then passed to another FC layer of size 16, followed by a softmax layer of size 2 to obtain the final prediction. We utilize the ReLu activation function for all other layers.

Experiments

Dataset and Ground-truth

Although we collected collusive users from the blackmarkets, it is unknown who among them are core blackmarket users. Thus, the ground-truth information about the core and compromised users are impossible to obtain unless blackmarkets themselves provide the data! We, therefore, consider the core and compromised users obtained from KORSE as the ground-truth since it uses the topological structure of the underlying collusive network to detect the core users. We hypothesize that KORSE is highly accurate in detecting core users. We also perform several case studies to validate our hypothesis. We intend to show how much NURSE (a non-topology based method) is close to KORSE (a pure topology-based method). We also present a case study to show whether the detected core users are really meaningful or not.

Since the number of compromised users (1,455) is 10 times higher than the number of core users (148), we generate two datasets for our analysis: **(i) Dataset (1:1)** is a balanced dataset where equal number of compromised users as that of core users are (randomly) sampled; **(ii) Complete dataset** is an imbalanced dataset where all collusive users are kept. We performed 10-fold stratified cross-validation and report the average performance.

Baseline Methods

Since ours is the first work to detect core blackmarket users, there is no existing baseline. We therefore design our own baselines by considering individual components of NURSE in isolation and their combinations:

1. **MFE:** This model uses only the metadata feature extractor.
2. **SFE:** This model uses only the similarity feature extractor.
3. **TFE:** This model uses only the textual feature extractor. Each comment is represented as a 786 dimensional vector using BERT.

We further combine these three components and design three more baselines: (4) **MFE+SFE**, (5) **MFE+TFE**, and (6) **SFE+TFE**.

These baselines also in turn serve the purpose of **feature ablation** to explain which features are important for NURSE.

Are core users the influential nodes in the network? To answer this, we consider three other approaches as baselines which aim to detect influential users:

7. **INF:** Huang et al. (2020) proposed a node influence indicator, called INF, based on the local neighboring information to detect influential nodes in a network.
8. **Weighted Betweenness Centrality (WBC):** Betweenness centrality (BC) (Brandes 2001) is a measure of node centrality based on the shortest paths. We utilize the approach in (Shin, Eliassi-Rad, and Faloutsos 2016) to run the weighted version of BC on CCN and detect core users.
9. **Coordination Game Model (CGM):** Zhang and Zhang (2017) proposed a coordination game model to find top-K nodes to maximize influence under certain spreading model.

Performance Comparison

Since all the competing methods return a score (or a probability), indicating the likelihood of a user being core, we first rank all the users based on the decreasing order of the score, and then measure the accuracy in terms of precision, recall, F1-Score and Area under the ROC curve (AUC) w.r.t. the 'core' class. Fig. 8 shows that NURSE dominates other baselines for almost all values of k (the top k users returned from the ranked list). Table 4 summarizes the performance (F1-Score and AUC) of the models at $k = 148$ (as there are 148 core users; it is also known as break even point) – NURSE turns out to be the best method, followed by MFE+SFE (for balanced dataset) and SFE+TFE (for imbalanced dataset).

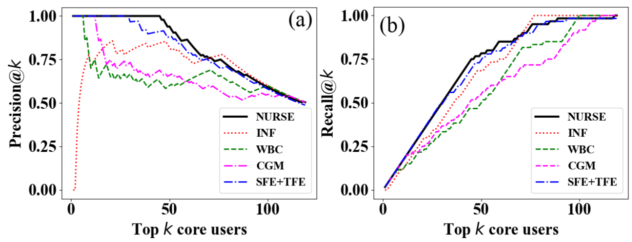


Figure 8: Change in the performance of competing methods with the increase of k (the number of results returned) for detecting core users from our dataset (1:1). For better visualization, among the variations of NURSE, we report the results of only the best variation (SFE+TFE).

Method	**Dataset** $(1:1)$		**Dataset** $(1:4)$	
	F1 (Core)	**AUC**	**F1 (Core)**	**AUC**
MFE	0.638	0.559	0.268	0.294
SFE	0.816	0.857	0.516	0.472
TFE	0.665	0.773	0.530	0.365
MFE+SFE	0.824	0.882	0.682	0.718
MFE + TFE	0.696	0.767	0.415	0.631
SFE+TFE	0.819	0.865	0.721	0.792
INF	0.750	0.139	0.533	0.113
WBC	0.617	0.304	0.407	0.270
CGM	0.622	0.392	0.302	0.414
NURSE	**0.879**	**0.928**	**0.833**	**0.845**

Table 4: Performance (F1-Score and AUC for detecting core users) of the competing methods at $k = 148$ (break-even point). The results also explain feature ablation of NURSE.

Similarity feature extractor (SFE) seems to be the most important component of NURSE, followed by TFE and MFE. Among influential node detection methods, both INF and CGM seem to be quite competitive. Next, we examine the core users identified by our proposed method KORSE and NURSE.

Case Studies

We further delve deeper into the characteristics of some of the core users detected by both KORSE and NURSE by conducting some case studies. These provide us strong evidences to validate our strategy of collecting the ground-truth from KORSE.

1. **Core users are heavy contributors**: A core user, on average, comments significantly ($\times 2.665$) more than a compromised user, indicating that core users are the top contributors to the freemium collusive market.
2. **Despite being heavy contributors, core users are not the largest beneficiaries of the collusive market**: We measure the average number of comments received by the videos uploaded by collusive users, and rank them in decreasing order of this quantity. We find only one core user from the top 30 users. Upon further investigation, we notice that only 8 out of the top 250 users are core users. This suggests that core users, despite being heavy contributors, are not the largest beneficiaries of the collusive market.
3. **Core users aggressively participate in the collusive market**: We observe that the average number of comments made per collusive video by core users is twice ($\times 1.997$) higher than that of compromised users. This indicates an aggressive behavior to promote the videos they comment on.
4. **Channels controlled by core users are not popular**: We observe that the channels controlled by core users are not the popular YouTube channels. More than 85% of the channels have a subscriber count of less than $1,000$. This clearly indicates that the primary objective of the core users is not to promote their own videos/channels.
5. **Channels controlled by core users have less uploaded videos**: We observe that the channels controlled by core users usually do not contain much YouTube videos. More than 90% of the channels have a video count of less than 100. This further corroborates the theory behind the working principle of core blackmarket users.

Despite the above suspicious characteristics exhibited by core channels, we observe that till date, 93% of the core channels **continue to be active on YouTube**. On average, these core channels have been active on YouTube for over 4 years (1497 days). It indicates how core channels are able to evade the current in-house fake detection algorithms deployed by YouTube.

Conclusion

This paper addressed the problem of detecting core users in YouTube blackmarkets. We curated a new dataset of collusive YouTube users. We then proposed KORSE, a novel graph-based method to segregate core users from compromised accounts. Empirical studies revealed interesting dynamics of core and compromised users. As KORSE is practically infeasible to design due to its dependency on the underlying collusive network, we further proposed NURSE, a deep fusion model that leverages only the user timeline and video submission information to detect core users. Extensive experiments on our dataset showed that NURSE is highly similar to KORSE in detecting core users. Summarizing, our study contributed in four aspects – problem definition, dataset, methods and empirical observation. As a future work, it would be interesting to see how NURSE can merge with existing collusive entity detection approaches to effectively identify core, collusive and non-collusive users. We also made the code and dataset publicly available.

Acknowledgements

T. Chakraborty would like to acknowledge the support of the Ramanujan Fellowship, ihub-Anubhuti-iiitd Foundation set up under the NM-ICPS scheme of the Department of Science and Technology, India, and Infosys Centre for AI at IIIT-Delhi.

References

Arora, U.; Dutta, H. S.; Joshi, B.; Chetan, A.; and Chakraborty, T. 2020. Analyzing and Detecting Collusive Users Involved in Blackmarket Retweeting Activities. *ACM TIST*, 11(3): 1–24.

Batagelj, V.; and Zaversnik, M. 2003. An O (m) algorithm for cores decomposition of networks. *arXiv preprint cs/0310049*.

Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R.; and Lefebvre, E. 2008. Fast unfolding of communities in large networks. *JSTAT*, 2008(10): P10008.

Borgatti, S. P.; and Everett, M. G. 2000. Models of core/periphery structures. *Social networks*, 21(4): 375–395.

Brandes, U. 2001. A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2): 163–177.

Bu, Z.; Xia, Z.; and Wang, J. 2013. A sock puppet detection algorithm on virtual spaces. *Knowledge-Based Systems*, 37: 366–377.

Castellini, J.; Poggioni, V.; and Sorbi, G. 2017. Fake twitter followers detection by denoising autoencoder. In *WI*, 195–202. ACM.

Cheng, J.; Ke, Y.; Chu, S.; and Özsu, M. T. 2011. Efficient core decomposition in massive networks. In *ICDM*, 51–62. IEEE.

Chetan, A.; Joshi, B.; Dutta, H. S.; and Chakraborty, T. 2019. Corerank: Ranking to detect users involved in blackmarket-based collusive retweeting activities. In *WSDM*, 330–338.

Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; and Tesconi, M. 2015. Fame for sale: Efficient detection of fake Twitter followers. *Decision Support Systems*, 80: 56–71.

Cucuringu, M.; Rombach, P.; Lee, S. H.; and Porter, M. A. 2016. Detection of core–periphery structure in networks using spectral methods and geodesic paths. *European Journal of Applied Mathematics*, 27(6): 846–887.

Della Rossa, F.; Dercole, F.; and Piccardi, C. 2013. Profiling core-periphery network structure by random walkers. *Scientific reports*, 3(1): 1–8.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dhawan, S.; Gangireddy, S. C. R.; Kumar, S.; and Chakraborty, T. 2019. Spotting Collective Behaviour of Online Frauds in Customer Reviews. In *IJCAI-19*, 245–251.

Dutta, H. S.; and Chakraborty, T. 2020. Blackmarket-Driven Collusion Among Retweeters–Analysis, Detection, and Characterization. *IEEE TIFS*, 15: 1935–1944.

Dutta, H. S.; and Chakraborty, T. 2020. Blackmarket-driven collusion on online media: a survey. *arXiv preprint arXiv:2008.13102*.

Dutta, H. S.; Chetan, A.; Joshi, B.; and Chakraborty, T. 2018. Retweet us, we will retweet you: Spotting collusive retweeters involved in blackmarket services. In *ASONAM*, 242–249. IEEE.

Dutta, H. S.; Dutta, V. R.; Adhikary, A.; and Chakraborty, T. 2020. HawkesEye: Detecting fake retweeters using Hawkes process and topic modeling. *IEEE Transactions on Information Forensics and Security*, 15: 2667–2678.

Dutta, H. S.; Jobanputra, M.; Negi, H.; and Chakraborty, T. 2021. Detecting and analyzing collusive entities on YouTube. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(5): 1–28.

Farooqi, S.; Zaffar, F.; Leontiadis, N.; and Shafiq, Z. 2017. Measuring and mitigating oauth access token abuse by collusion networks. In *IMC*, 355–368.

Hooi, B.; Shin, K.; Lamba, H.; and Faloutsos, C. 2020. TellTail: Fast Scoring and Detection of Dense Subgraphs. In *AAAI*, 4150–4157.

Huang, X.; Chen, D.; Wang, D.; and Ren, T. 2020. Identifying Influencers in Social Networks. *Entropy*, 22(4): 450.

Kojaku, S.; and Masuda, N. 2017. Finding multiple core-periphery pairs in networks. *Physical Review E*, 96(5): 052313.

Lettry, L.; Perdoch, M.; Vanhoey, K.; and Van Gool, L. 2017. Repeated pattern detection using CNN activations. In *WACV*, 47–55.

Malliaros, F. D.; Papadopoulos, A. N.; and Vazirgiannis, M. 2016. Core Decomposition in Graphs: Concepts, Algorithms and Applications. In *EDBT*, 720–721.

Mislove, A.; Marcon, M.; Gummadi, K. P.; Druschel, P.; and Bhattacharjee, B. 2007. Measurement and analysis of online social networks. In *SIGCOMM*, 29–42.

Peng, C.; Kolda, T. G.; and Pinar, A. 2014. Accelerating community detection by using k-core subgraphs. *arXiv preprint arXiv:1403.2226*.

Rombach, M. P.; Porter, M. A.; Fowler, J. H.; and Mucha, P. J. 2014. Core-periphery structure in networks. *SIAM Journal on Applied mathematics*, 74(1): 167–190.

Shah, N.; Lamba, H.; Beutel, A.; and Faloutsos, C. 2017. The many faces of link fraud. In *ICDM*, 1069–1074. IEEE.

Shin, K.; Eliassi-Rad, T.; and Faloutsos, C. 2016. Corescope: Graph mining using k-core analysis—patterns, anomalies and algorithms. In *ICDM*, 469–478. IEEE.

Shu, K.; Wang, S.; and Liu, H. 2018. Understanding user profiles on social media for fake news detection. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 430–435. IEEE.

Xiang, B.-B.; Bao, Z.-K.; Ma, C.; Zhang, X.; Chen, H.-S.; and Zhang, H.-F. 2018. A unified method of detecting core-periphery structure and community structure in networks. *Chaos*, 28(1): 013122.

Zhang, F.; Zhang, Y.; Qin, L.; Zhang, W.; and Lin, X. 2016. When engagement meets similarity: efficient (k, r)-core computation on social networks. *arXiv preprint arXiv:1611.03254*.

Zhang, F.; Zhang, Y.; Qin, L.; Zhang, W.; and Lin, X. 2017. Finding critical users for social network engagement: The collapsed k-core problem. In *AAAI*.

Zhang, Y.; and Zhang, Y. 2017. Top-K influential nodes in social networks: A game perspective. In *SIGIR*, 1029–1032.