Interpretable Cross-Platform Coordination Detection on Social Networks

Auriant Emeric[1(✉)] and Chomel Victor[2]

[1] École Polytechnique, Palaiseau, France
emeric.auriant@polytechnique.edu
[2] Institut des Systèmes Complexes Paris Ile de France (ISCPIF) - CNRS, Paris, France
victor.chomel@ispcif.fr

Abstract. Numerous disinformation campaigns are operating on social networks to influence public opinion. Detecting these campaigns primarily involves identifying coordinated communities. As disinformation campaigns can take place on several social networks at the same time, the detection must be cross-platform to get a proper picture of it. To encode the different types of coordination, a multi-layer network is built. We propose a scalable coordination detection algorithm, adapted from the Louvain algorithm and the Iterative Probabilistic Voting Consensus algorithm. This algorithm is applied to the previously built multi-layer network. Users from different social networks are then embedded in a common space to link communities with similar interests. This paper introduces an interpretable and modular framework used on three datasets to prove its effectiveness for coordination detection method and to illustrate it with real examples.

Keywords: Coordination detection · User Alignment · Social Networks

1 Introduction

Although disinformation campaigns always existed, they are now ubiquitous on online social networks (OSN). Disinformation is understood here as purposefully spreading misleading or inaccurate information in order to influence public opinion. These campaigns use various techniques such as creating deepfakes, fake news, or astroturfing. Some examples include the 2016 US presidential election [8], the COVID-19 pandemic [25], or the recent Spamouflage Operation [3].

Among disinformation tactics, the focus here is on Coordinated Inauthentic Behaviours (CIB) as defined by Meta [1]. It is based on using multiple social media accounts to mislead the online debate [25]. These campaigns are seen as a threat to freedom of speech and the war on them is intensifying. OSNs, such as Facebook, are at least since 2018 studying how to prevent them and are regularly

H. Cherifi et al. (Eds.): COMPLEX NETWORKS 2023, SCI 1144, pp. 143–155, 2024.
https://doi.org/10.1007/978-3-031-53503-1_12

deleting troublesome accounts. Between 2017 and 2021, Facebook identified 17 billion fake accounts [23]. Similarly, independent and state-run organisations, such as the EU Disinfo Lab, are fighting against disinformation by detecting these campaigns and alerting the relevant authorities [2].

However, CIB detection is not an easy task. Users taking part in such campaigns may be split into different OSNs [17,36], which complicates the detection of the whole operation. In addition, coordination and inauthenticity are very different concepts. Coordination can be defined as the organisation, intentional or not, of actors to achieve common goals. On the other hand, a user is considered inauthentic if his behaviour falls outside the norms observed on the network: for instance troll farms or bots. Consequently, these notions can be unrelated: activists' accounts on OSNs can spread disinformation on the same subject without being coordinated, while a group of users can have a coordinated but authentic behaviour, as coordination arises naturally from shared interests or opinions. The issue of authenticity is not directly discussed here as the focus is on coordination detection which is a mandatory first step in the process of CIB detection. When needed, the inauthenticity is assessed manually but this question has already been addressed in similar cases [29,34].

Contributions. In this article, a technical framework is proposed to detect coordinated communities spanning across multiple OSNs. First, a multi-layer network adapted to coordination detection is built from the collected interactions between users. This network encodes the various behaviours and types of coordination through graphs of interactions with various time thresholds. Then, a scalable community detection algorithm, adapted from a consensus clustering algorithm, is applied to this multi-layer network to get communities. Finally, similar users from different OSNs are found by embedding them in a shared latent space without using prior knowledge of identical users on several OSNs. These users can then be used to link communities with shared interests.

2 Related Work

The detection of CIB is an understudied topic from an academic point of view but may overlap with bot detection [18] or fake-news detection [37], including multimodal and cross-platform methods [22]. Generative models can be used to generate embeddings used for coordination detection [33] but are not easily interpretable. A common way of doing such detection while keeping the interpretability is to look at content propagation and to study graphs of interactions. In this context, an OSN can be represented as a multi-layered network [28], a widely used object [11]. Each action, such as mention or quote, is considered separately to build interaction or co-interaction graphs [28,34]. In interaction graphs, the edges represent the actions and are directed from the user to the object of the action. In co-interaction graphs, the nodes represent the users and two nodes are linked if their users performed the same action. The temporal dimension of the co-interaction can be encoded in the edges of the graph. To

this end, time thresholds are introduced. Two nodes are neighbours if the users did the same action within a limited time period [34].

Various methods have been studied to detect communities on a multi-layer graph. The simplest would be using a single layer, such as the co-retweet layer [14], collapsing the multi-layer graph to a simple graph by summing the weight of the edges [35] or creating a similarity graph using all the layers [4,15]. Classical community detection algorithms such as the Louvain algorithm [7] can then be used to detect communities [24]. These methods are straightforward but involve a loss of information. To avoid this, community detection algorithms adapted to multi-layer graphs can also be used but are often adapted to small graphs [16].

Features, such as the mean degree or the main K-Core number for each layer, can be extracted for each community. Embeddings from deep learning models could also be used to replace or to enrich feature vectors. These vectors are finally used to train a classifier to detect coordination [29,34].

Coordinated campaigns are not limited to a single platform [17,36]. To the best of our knowledge, cross-platform coordination detection methods exist but their results cannot be easily interpreted [38]. In order to go further and perform interpretable multi-platform community detection, the various multi-layer networks of each platform need to be aligned. To do so, a cross-platform network user identification is done, which means finding users with the same identity on different social networks. Digital footprints such as username, description, profile picture, or even location can be used to find some correspondences [21]. Another method is to perform graph alignment by using pre-aligned users across the networks, called anchors, and deep learning methods [13,19,20]. These aligned users can then be used to find similar communities from different social networks.

3 Dataset

Community Detection Benchmark. A first dataset called *politic-ie* [15], was used to evaluate the multi-layer community detection algorithm. This dataset is a 9-layer network created with 267K tweets from 348 Irish politicians and political organisations split into 7 communities according to their political affiliation.

Cross-platform Dataset. Two datasets from different OSNs are used to study community alignment. The first is the Pushshift telegram dataset [5]. This dataset consists of 317M messages sent on 27,801 channels by 2.2M users between 2015 and 2019. The second is a Twitter dataset [12], which is composed of 88M tweets from 150K users. This dataset was used as it is one of the largest directly available, regarding the recent restrictions on Twitter dataset availability.

Both datasets contain messages on a wide range of topics. In order to have comparable data volume, only messages sent in January 2019 are used. In addition, on Telegram, messages from chats are ignored, only messages from public channels are kept. Messages without text and links are removed; these messages often contained photos or videos that were not included in the original dataset. After this filtering step, the Twitter dataset contains 421K messages from 32K users and the Telegram dataset contains 624K messages from 9K users.

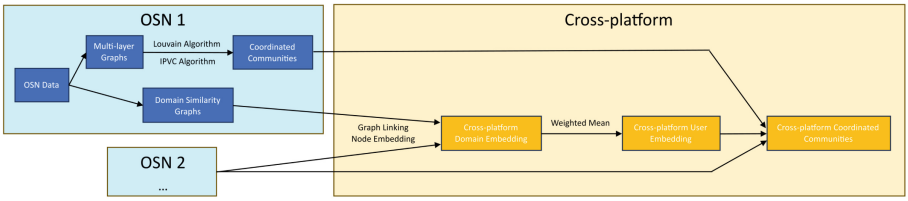

Fig. 1. Overview of the proposed framework in the case of two OSNs

Ukraine War Dataset. To present an application of the method, the algorithms were tested on another dataset related to an ongoing event. This dataset is composed of about 1M messages from three different OSNs: Twitter (440K Tweets), Telegram (424K messages), and VK (191K). This dataset is built from messages containing a list of keywords related to the Ukraine War posted in May and June 2023 and is composed mainly of messages in English and Russian.

4 Method

The method consists of two independent algorithms applied successively and explained in the following sections. The results of both algorithms are finally used to link communities with shared interests.

4.1 Multi-layer Network Community Detection

Multi-layer Network. Through OSNs, users can interact with each other by performing a variety of actions such as posting messages, hashtags, or even links. Here, an OSN is represented as a multi-layer network. Each layer is an undirected graph corresponding to an interaction and a time threshold δ. Two nodes are linked if they did the same action within δ seconds from each other. The studied interactions are the following: co-message, co-share (for instance, co-retweet on Twitter), co-hashtag, co-URL and co-domain. Two layers, the co-message and co-domain layers are special. The co-message layer is based on an embedding of the message obtained with the Sentence-Bert Model [30]. In this layer, two nodes are linked if their embeddings have a very high cosine similarity value. This layer is used to detect extremely similar content, such as copy and paste with few modifications. The co-domain layer uses the domain name of the URLs instead of the full URLs and can therefore detect users following similar media. The weight of the edges is the natural logarithm of the number of co-interactions within the time threshold between the two accounts. This reweighting avoids having extremely heavy edges that could interfere with the general clustering.

The layers are filtered to remove natural interactions between users. First, co-interactions too often performed are ignored: retweeting a mainstream article does not provide useful information about coordination as it is widely shared.

Then, the lowest weighted edges are removed from each layer to suppress weak or spurious connections. To avoid using an arbitrary weight threshold, a fixed percentage of edges is removed from each layer. Finally, nodes of degree zero are removed from each layer. As a result, many nodes are not present on all layers.

Community Detection. The final clustering is performed on all the nodes of the multi-layer network, i.e. the union of the nodes of all layers. Given the large number of nodes, the community detection algorithm needs to be fast and memory efficient so each layer is first processed independently using the Louvain algorithm [7]. For each layer l, it returns of partition function π_l such that, for a node x, $\pi_l(x) = k$ means that, in the layer l the node x is in the cluster k. Then, an Iterative Probabilistic Voting Consensus (IPVC) algorithm is used [26]. This algorithm aims at finding a clustering, called consensus clustering, by minimising the average probability that two nodes do not belong to the same community in a layer if they are from the same community in the consensus clustering (see 1).

$$\pi^*(x) = \arg \min_{i \in [\![1:m]\!]} \sum_{l=1}^{L} \mathbb{P}\left(\pi_l(x) \neq \pi_l(x') | x' \in \mathbb{X}, \pi^*(x') = i\right) \tag{1}$$

with π^* the consensus clustering, m the number of clusters, L the number of layers and X the set of nodes. This method aims at retaining as much information from each layer as possible while being adapted to a large amount of data.

As layers do not have the same importance to detect coordination, the arithmetic mean of the probabilities for each layer can be weighted. Weights are selected as the values of the first singular vector of the distance matrix between the clusterings of each layer. A constant based on the minimum value is finally added to ensure that all weights are strictly positive. The distance used is the Network Partition Distance [9], which corresponds to the fraction of nodes that best-matching communities in two layers do not have in common. This matrix shows which layers provide different coordination information. Its first singular vector sums up the relations between the layers. A layer containing information different from the others will have a higher weight and thus a greater impact.

4.2 Cross-Platform Community Alignment

The second step is to match up similar communities. Here, communities are considered similar if they have similar users.

Similarity Graph. The method used is inspired by the Sim-Clusters algorithm [31]. For each network, a bipartite graph composed of nodes representing accounts and domain names is created. The weight of an edge between an account and a domain name corresponds to the number of URLs containing the domain name posted by the account. Nodes representing domains that are too common or that do not carry information, for example, URL shorteners or OSNs (such as twitter.com), are removed from the graph. Finally, the edges are normalised so that each node representing a user is of degree one.

Then the domain similarity graph is computed. This graph is a weighted undirected graph whose nodes are domain names. The edges are weighted according to the cosine similarity between the columns of the adjacency matrix of the bipartite graph. Edges with weight inferior to a threshold are discarded to select meaningful links and remove noise and spurious connections.

Common Domain Latent Space. Users on various OSNs often post links from identical domains, which means common nodes in similarity graphs. The set of pairs of nodes representing the same domain name on different OSNs is noted $\overline{N}$. These nodes are used as anchors to compute a common latent space for all similarity graphs. To do so, similarity graphs of each OSN are linked by adding edges between pairs of nodes, leading to a cross-platform similarity graph.

An embedding of the nodes of this cross-platform similarity graph is then computed using Spectral Embedding [6]. The embedding of a node n is noted $\mathbf{e}_n$. This relatively simple embedding allows discrimination of nodes present in different connected components and does not involve any prior training. As the pairs of nodes of $\overline{N}$, represent the same domain name on different OSNs, their embedding must be highly similar. To maximise this similarity without altering the one between embeddings in a given OSN, the embeddings of each OSN are multiplied by an orthogonal matrix $\mathbf{O}^*$, computed by solving the Orthogonal Procruste Problem for the embeddings of the anchor nodes [32]:

$$\mathbf{O}^* = \arg \min_{\mathbf{O}\mathbf{O}^T=I} \sum_{(n,m)\in\overline{N}} ||\mathbf{O}\mathbf{e}_n - \mathbf{e}_m||_2 \tag{2}$$

User Embedding. The user embeddings are defined as the frequency-weighted average of the embedding of the domain names they posted. These embeddings are only computed for users who posted more than a minimal number of messages with links. The embedding of users with too few messages might be of poor quality if these messages do not reflect their usual behaviour.

Community Linking. Pairs of users with high similarity across OSNs, or from the same OSN can be created. These pairs are then used to quantify the proximity between the communities previously detected. The similarity measure between two communities is measured with the proportion of users paired in each community and the number of pairs over the number of users paired. These metrics illustrate the density of connections between the communities.

4.3 Overview of the Framework

An overview of the framework for two OSNs is shown in Fig. 1. First, for each OSN a multi-layer network is created from the various co-interaction. Then community detection is performed on each network on each layer and combined using the IPVC algorithm. Then the similarity graphs, defined Sect. 4.2, are also created for each OSN. These graphs are linked through anchor nodes to create a

cross-platform similarity graph which is then embedded using Spectral Embedding. The similarity between the embeddings of anchor nodes is maximised by using an orthogonal transformation. An embedding of the users is then obtained by computing the average of the embeddings of the domain names. Finally, communities sharing users with highly similar embeddings are linked.

Table 1. Clustering results with 1000 simulations on the politic-ie dataset

Clustering method	AMIS	ARS
Single (best layer)	0.835 ± 0.016	0.891 ± 0.012
Single (worst layer)	0.035 ± 0.025	0.034 ± 0.024
Collapsed	0.761 ± 0.053	0.770 ± 0.092
Similarity	$\mathbf{0.859 \pm 0.008}$	0.893 ± 0.003
Consensus	0.852 ± 0.025	$\mathbf{0.908 \pm 0.038}$

5 Results

Community Detection Benchmark. The method presented in Sect. 4.1 was used on the dataset *politic-ie*. To assess the performance of the method, two measures were computed on the detected communities: the Adjusted Mutual Info Score (AMIS) and the Adjusted Rand Score (ARS) [10]. These two scores are commonly used to measure agreement between partitions, and are here adjusted (the expected value is zero when the partitions are made at random). Other community detection methods were tested such as: using the Louvain algorithm on a single layer, collapsing the network by summing edges' weights of each layer before using the Louvain algorithm, or computing a similarity graph [4].

Each method was applied 1000 times to get means and standard deviations (Fig. 1). The quality of the community detection using a single layer is highly variable, proving the benefits of using a multi-layered approach. The method using the collapsed graph is worse than the method with the best layer but easier to use, as there is no need to choose a layer. Similarity and consensus methods provide the best results and their performances are similar and depend on studied metrics. Our consensus method can thus be used on multi-layer networks.

Multi-layer Network. In this paragraph and the next two, the dataset used is the *Cross-platform Dataset*. Various layers are presented in Fig. 2: two graphs corresponding to co-domain, respectively with a time threshold of an hour and a day, the third graph represents co-mention with a time threshold of a day. As expected, the co-interaction graph with a time threshold of an hour has a lower density than the others. Fewer interactions occur within a time threshold of an hour than within the course of a day. The information on this graph is therefore important, it indicates more coordinated users. However, this graph

also contains fewer nodes, the other graphs are needed to identify the complete communities. The graphs of co-mention and co-domain also help in this way. By combining these graphs, more finely tuned communities can be identified.

Fig. 2. Various graphs of co-interaction computed on the Twitter dataset of the *Cross-platform dataset*. The node colours correspond to communities obtained with the Louvain algorithm. **Left.** Co-domain in an hour **Middle.** Co-domain in a day **Right.** Co-mention in a day

Cross-platform Domain Name Embedding. Computing the embeddings is the next step. To have significant similarity measures between domain names, only names posted more than 10 times in our datasets are used to create the similarity graphs. This results in a graph with 1048 nodes for Twitter and 584 nodes for Telegram. These two graphs are then linked using 105 pairs of anchor nodes before doing a spectral embedding. After the orthogonal rotation, the mean similarity value between the two embeddings of an anchor is 0.999 and the minimal similarity is 0.992. A t-SNE of the embeddings is presented in Fig. 3. In addition to anchors embeddings, other users' embeddings from different OSNs appear to be highly similar, some examples can be seen at the top of Fig. 3. A post-hoc study shows that these nodes represent users with shared interests which validates the approach. In some places, for example the left of Fig. 3, only nodes from one OSN are present[1]. This can be explained by the fact that the topics covered by the dataset do not totally overlap.

To ensure that the embeddings are meaningful, the domain name extensions are studied. The embeddings corresponding to the 10 most frequent domain name extensions (except .com, .org, and .net), are shown in Fig. 3. The embeddings appear to be gathered by extension, for example at the bottom with the .ir extension or at the top with the .de extension. They are therefore relevant, as they enable identification of the geographical origin of a website. This gives us good hope that both graphs are embedded in a common meaningful space.

Community Linking. Finally, communities from the two OSNs can be matched using the metrics detailed paragraph 4.2. The community on Telegram speaks

[1] It should be noted that once communities have been matched, OSNs communities structures can be studied in detail for other purposes.

Russian, while the Twitter one speaks English. Both communities discuss about streaming video games and PlayStations. These communities share common interests which are detected despite language differences.

Two metrics are used to assess the quality of matching. Messages related to co-interactions in each community are extracted and embedded using the Sentence-Bert model [30]. This embedding is then used to compute cosine similarities between communities. The average similarity between this gamer Telegram community and other communities on Twitter is 0.02 (with s.d. of 0.08), meanwhile, the similarity between the two matched communities is 0.44.

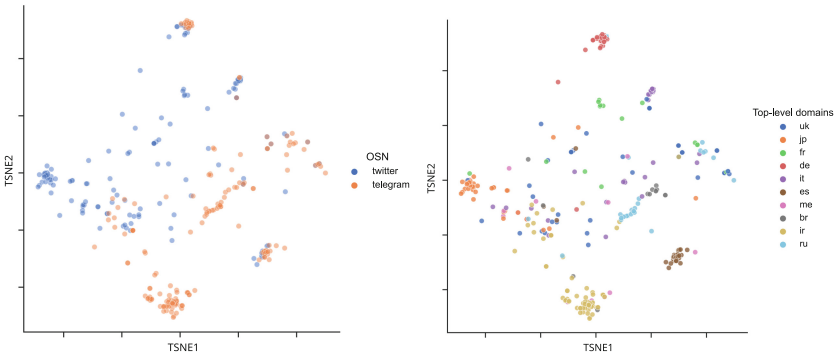

Fig. 3. t-SNE of the domain names embeddings obtained with two OSNs on the *Cross-platform Dataset*. The metric used is the cosine similarity. Only domain name embeddings whose extension is among the 10 most frequent are kept. **Left.** OSNs of origin of the domain name. **Right.** Extension of the domain name

A similarity between two users can also be defined as the maximal cosine similarity between the messages of these users. This user similarity can then be averaged to obtain a similarity between communities. The community similarity between the two linked communities is 0.215 meanwhile the average similarity between one of the linked communities and the communities of the other OSN is 0.072 (with s.d. of 0.047).

These metrics confirm that the heuristic used to match communities is meaningful from a semantic point of view but the alignment performed goes further than this. Users, and communities, that are brought together are those sharing information sources and therefore narratives, hence the semantic similarity.

Detected Coordination Example. The framework was also applied to the *Ukraine War Dataset*. All the layers presented in Subsect. 4.1, or their equivalent on the corresponding OSN, coupled with three time thresholds (a minute, an hour, and a day) were used to create the multi-layer network. On each OSN, several types of coordinated communities were detected such as:

– Bot-like users sharing a similar narrative at the same time. This community creates a clique in the co-message layer with a threshold of a minute. The

coordination is confirmed by the near-identical user description. These facts suggest a communication agency or a troll farm.
- A community of Russian embassy accounts and unknown users sharing pro-Russian content. In this case, the coordination appears because of the heavy edges on the various graphs with thresholds of an hour or even a day.
- Users whose only activity is to relay articles published by a given newspaper. These users are neighbours in the co-URL layers.

The embedding of domains and users also helps identify useful connections. For example, in the case of domain embeddings, American conspiracy websites have highly similar embeddings. Hence, at the user level, people who regularly promote these media have similar embeddings. Linking the coordinated communities to which these users belong, brings together communities with common narratives. When these narratives are part of disinformation campaigns, it is legitimate to assume that these linked communities are coordinated. Another interesting example is the fact that the embeddings of journalists are very similar to the embeddings of the newspaper they are working for. These embeddings therefore connect the different accounts of the newspapers and their journalists on the different OSNs. In this case, the coordination is obvious.

6 Discussion

In this article, the process of creating a multi-layer network encoding the various types of coordination was presented. Communities detected on various OSNs were then matched using domain similarity graphs built from the links posted by the users. Our clustering method was proven to be reliable on a labelled dataset. Then, two cross-platform datasets were used to illustrate the coordination detection and the community matching with examples. Finally, various metrics were introduced to demonstrate the effectiveness of the method.

Different types of coordination have been detected between journalists, official entities, or users suspected of being bots. In addition, coordinated communities on different networks have been brought together as they post links from similar websites. The examples shown were players from different countries who spoke different languages, conspiracy theorists, or journalists and their newspapers. Once communities have been linked, the various co-interactions between the users and their temporality can easily be observed on the multi-layer graph. This observation enables an analyst to identify the type of coordination and thus assess its authenticity.

However, this method does not detect the source of coordination. For example, if a user asks others to retweet him. The coordination related to the retweets can be detected but the user who posted the initial tweet will not be included in the community. Moreover, this method does not detect recurrent behaviours or co-interaction with a time span greater than the time threshold. To solve these problems, layers could be added to the multi-layer network, at the expense of temporal and spatial complexities of the method.

As stated in Sect. 2, once communities have been detected, features can be extracted to get an embedding per community. These embeddings have then been used, in other articles, to discriminate the communities into two categories: coordinated or uncoordinated communities [34]. This classification would complete the CIB detection process. Furthermore, training an explainable classifier, such as an Explainable Boosting Model [27], would allow to have a better understanding of the characteristic and of how these campaigns work.

Finally, the proposed framework is extremely modular. The network layers, clustering and embedding algorithms or the community matching methods can be freely modified to adapt the method to the available resources.

References

1. Coordinated inauthentic behavior explained. https://about.fb.com/news/2018/12/inside-feed-coordinated-inauthentic-behavior/
2. Doppelganger - media clones serving russian propaganda. https://www.disinfo.eu/doppelganger/
3. Raising online defenses through transparency and collaboration. https://about.fb.com/news/2023/08/raising-online-defenses/
4. Alimadadi, F., Khadangi, E., Bagheri, A.: Community detection in facebook activity networks and presenting a new multilayer label propagation algorithm for community detection **33**(10), 089 (1950). https://doi.org/10.1142/S0217979219500899
5. Baumgartner, J., Zannettou, S., Squire, M., Blackburn, J.: The pushshift telegram dataset
6. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation **15**(6), 1373–1396. https://doi.org/10.1162/089976603321780317
7. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks **2008**(10), P10,008. https://doi.org/10.1088/1742-5468/2008/10/P10008
8. Bovet, A., Makse, H.A.: Influence of fake news in twitter during the 2016 US presidential election **10**(1), 7. https://doi.org/10.1038/s41467-018-07761-2
9. Calatayud, J., Bernardo-Madrid, R., Neuman, M., Rojas, A., Rosvall, M.: Exploring the solution landscape enables more reliable network community detection **100**(5), 052308. https://doi.org/10.1103/PhysRevE.100.052308
10. Chacón, J.E., Rastrojo, A.I.: Minimum adjusted rand index for two clusterings of a given size **17**(1), 125–133. https://doi.org/10.1007/s11634-022-00491-w
11. De Domenico, M.: More is different in real-world multilayer networks. https://doi.org/10.1038/s41567-023-02132-1
12. Enryu: Fun with large-scale tweet analysis. https://medium.com/@enryu9000/fun-with-large-scale-tweet-analysis-783c96b45df4
13. Gao, S., Zhang, Z., Su, S., Yu, P.S.: REBORN: transfer learning based social network alignment **589**, 265–282. https://doi.org/10.1016/j.ins.2021.12.081
14. Graham, T., Bruns, A., Zhu, G., Campbell, R.: Like a virus: the coordinated spread of coronavirus disinformation
15. Greene, D., Cunningham, P.: Producing a unified graph representation from multiple social network views
16. Huang, X., Chen, D., Ren, T., Wang, D.: A survey of community detection methods in multilayer networks **35**(1), 1–45. https://doi.org/10.1007/s10618-020-00716-6

17. Jakesch, M., Garimella, K., Eckles, D., Naaman, M.: Trend alert: How a cross-platform organization manipulated twitter trends in the indian general election **5**, 1–19. https://doi.org/10.1145/3479523
18. Kudugunta, S., Ferrara, E.: Deep neural networks for bot detection **467**, 312–322. https://doi.org/10.1016/j.ins.2018.08.019
19. Lei, T., Ji, L., Wang, G., Liu, S., Wu, L., Pan, F.: Transformer-based user alignment model across social networks **12**(7), 1686. https://doi.org/10.3390/electronics12071686
20. Liu, L., Zhang, Y., Fu, S., Zhong, F., Hu, J., Zhang, P.: ABNE: an attention-based network embedding for user alignment across social networks **7**, 23,595–23,605. https://doi.org/10.1109/ACCESS.2019.2900095
21. Malhotra, A., Totti, L., Meira Jr., W., Kumaraguru, P., Almeida, V.: Studying user footprints in different online social networks. https://doi.org/10.48550/arXiv.1301.6870
22. Micallef, N., Sandoval-Castañeda, M., Cohen, A., Ahamad, M., Kumar, S., Memon, N.: Cross-platform multimodal misinformation: Taxonomy, characteristics and detection for textual posts and videos **16**, 651–662. https://doi.org/10.1609/icwsm.v16i1.19323
23. Moore, M.: Fake accounts on social media, epistemic uncertainty and the need for an independent auditing of accounts. https://doi.org/10.14763/2023.1.1680
24. Morstatter, F., Shao, Y., Galstyan, A., Karunasekera, S.: From *Alt-Right to Alt-Rechts*: Twitter analysis of the 2017 german federal election. In: Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18, pp. 621–628. ACM Press. https://doi.org/10.1145/3184558.3188733
25. Murero, M.: Coordinated inauthentic behavior: An innovative manipulation tactic to amplify COVID-19 anti-vaccine communication outreach via social media **8**, 1141416. https://doi.org/10.3389/fsoc.2023.1141416
26. Nguyen, N., Caruana, R.: Consensus clusterings. In: Seventh IEEE International Conference on Data Mining (ICDM 2007), pp. 607–612. https://doi.org/10.1109/ICDM.2007.73. ISSN: 2374-8486
27. Nori, H., Jenkins, S., Koch, P., Caruana, R.: InterpretML: A unified framework for machine learning interpretability
28. Pierri, F., Artoni, A., Ceri, S.: HoaxItaly: a collection of italian disinformation and fact-checking stories shared on twitter in 2019. https://doi.org/10.48550/arXiv.2001.10926
29. Pierri, F., Piccardi, C., Ceri, S.: A multi-layer approach to disinformation detection in US and italian news spreading on twitter **9**(1), 1–17. https://doi.org/10.1140/epjds/s13688-020-00253-8
30. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using siamese BERT-networks. https://doi.org/10.48550/arXiv.1908.10084
31. Satuluri, V., et al.: SimClusters: Community-based representations for heterogeneous recommendations at twitter. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '20, pp. 3183–3193. Association for Computing Machinery. https://doi.org/10.1145/3394486.3403370
32. Schönemann, P.H.: A generalized solution of the orthogonal procrustes problem **31**(1), 1–10. https://doi.org/10.1007/BF02289451
33. Sharma, K., Zhang, Y., Ferrara, E., Liu, Y.: Identifying coordinated accounts on social media through hidden influence and group behaviours. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 1441–1451. ACM. https://doi.org/10.1145/3447548.3467391

34. Vargas, L., Emami, P., Traynor, P.: On the detection of disinformation campaign activity with network analysis. https://doi.org/10.48550/arXiv.2005.13466
35. Weber, D., Neumann, F.: Amplifying influence through coordinated behaviour in social networks **11**(1), 111. https://doi.org/10.1007/s13278-021-00815-2
36. Wilson, T., Starbird, K.: Cross-platform disinformation campaigns: Lessons learned and next steps **1**(1). https://doi.org/10.37016/mr-2020-002
37. Zhang, C., Gupta, A., Kauten, C., Deokar, A.V., Qin, X.: Detecting fake news for reducing misinformation risks using analytics approaches **279**(3), 1036–1052. https://doi.org/10.1016/j.ejor.2019.06.022
38. Zhang, Y., Sharma, K., Liu, Y.: Capturing cross-platform interaction for identifying coordinated accounts of misinformation campaigns. In: Kamps, J., et al. (eds.) Advances in Information Retrieval, Lecture Notes in Computer Science, pp. 694–702. Springer, Heidelberg (2023). https://doi.org/10.1007/978-3-031-28238-6_61