Closeness Centrality Detection in Homogeneous Multilayer Networks

Hamza Reza Pavel, Anamitra Roy, Abhishek Santra and Sharma Chakravarthy
Computer Science and Engineering Department and IT Lab, UT Arlington, U.S.A.

Keywords: Homogeneous Multilayer Networks, Closeness Centrality, Decoupling Approach, Accuracy & Precision.

Abstract: Centrality measures for simple graphs are well-defined and several main-memory algorithms exist for each. Simple graphs have been shown to be not adequate for modeling complex data sets with multiple types of entities and relationships. Although multilayer networks (or MLNs) have been shown to be better suited, there are very few algorithms for centrality measure computation *directly* on MLNs. Typically, they are converted (aggregated or projected) to simple graphs using Boolean AND or OR operators to compute various centrality measures, which is *not only inefficient but incurs a loss of structure and semantics*.
In this paper, algorithms have been proposed that compute closeness centrality on an MLN *directly* using a novel decoupling-based approach. Individual results of layers (or simple graphs) of an MLN are used and a composition function is developed to compute the closeness centrality nodes for the MLN. The challenge is to do this efficiently while preserving the accuracy of results with respect to the ground truth. However, since these algorithms use only layer information and do not have complete information of the MLN, computing a global measure such as closeness centrality is a challenge. Hence, these algorithms rely on heuristics derived from intuition. The advantage is that this approach lends itself to parallelism and is more efficient than the traditional approach. Two heuristics, termed CC1 and CC2, have been presented for composition and their accuracy and efficiency have been empirically validated on a large number of synthetic and real-world-like graphs with diverse characteristics. CC1 is prone to generate false negatives whereas CC2 reduces them, is more efficient, and improves accuracy.

1 INTRODUCTION

Closeness centrality measure, a global graph characteristic, defines the importance of a node in a graph with respect to its distance from all other nodes. Different centrality measures have been defined, both local and global, such as degree (Bródka et al., 2011), closeness (Cohen et al., 2014), eigenvector (Solá et al., 2013), multiple stress (Shi and Zhang, 2011), betweenness (Brandes, 2001), harmonic (Boldi and Vigna, 2014), and PageRank (Pedroche et al., 2016). Closeness centrality defines the importance of a node based on *how close it is to all other nodes* in the graph. Closeness centrality can be used to identify nodes from which communication with *all other nodes in the network* can be accomplished in least number of hops. Most of the centrality measures are defined for simple graphs or monographs. Only page rank centrality has been extended to multilayer networks (De Domenico et al., 2015; Halu et al., 2013).

A multilayer network consists of layers, where each layer is a simple graph consisting of nodes (entities) and edges (relationships) and optionally connected to other layers through inter-layer edges. If one were to model the three social media networks Facebook, LinkedIn, and Twitter, an MLN is a better model as there are multiple edges (connections) between any two nodes (see Fig. 1.) This type of MLN is

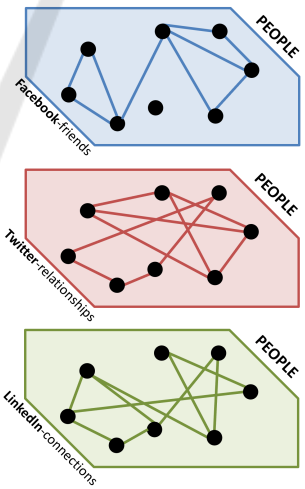


Figure 1: HoMLN Example.

Pavel, H., Roy, A., Santra, A. and Chakravarthy, S.
Closeness Centrality Detection in Homogeneous Multilayer Networks.
DOI: 10.5220/0012159500003598
In *Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2023) - Volume 1: KDIR*, pages 17-29
ISBN: 978-989-758-671-2; ISSN: 2184-3228

categorized as homogeneous (or HoMLNs) as the set of entities in each layer has a common subset, but relationships in each layer are different. It is also possible to have MLNs where each layer has different types of entities and relationships within and across layers. Modeling the DBLP data set (dbl, 1993) with authors, papers, and conferences needs this type of heterogeneous MLNs (or HeMLNs) (Kivelä et al., 2014).

This paper presents two heuristic-based algorithms to compute closeness centrality nodes (or CC nodes) on HoMLNs with high accuracy and efficiency. The challenge is in computing a global measure using partitioned graphs (layers in this case) and composing them with minimal additional information to compute a global measure for the combined layers. Boolean AND composition of layers is used for ground truth in this paper (OR is another alternative). These MLN algorithms use the decoupling-based approach proposed in (Santra et al., 2017a; Santra et al., 2017b). Based on this approach, closeness centrality is computed on each layer *once* and minimal additional information is kept from each layer to compose. With this, one can *efficiently* estimate the CC nodes of the MLN. This approach has been shown to be application-independent, efficient, lends itself to parallel processing (of layers), and is flexible to compute centrality measure on any arbitrary subset of layers.

Problem Statement: Given a homogeneous MLN (HoMLN) with l number of layers – G_1 *(V,* E_1*),* G_2 *(V,* E_2*), ...,* G_l *(V,* E_l*)*, where V and E_i are the vertex and edge set in the i^{th} layer – the goal is to identify the closeness centrality nodes of the Boolean AND-aggregated layer consisting of any r layers using the partial results obtained from each of the layers during the analysis step where $r \leq l$. In the decoupling-based approach, the analysis function is defined as the Ψ step, and the closeness centrality nodes are estimated in the composition step or the Θ step using the partial results obtained during the Ψ step.

1.1 Contributions and Paper Outline

Contributions of this paper are:

- Algorithms for computing closeness centrality nodes of MLNs
- Two heuristics to improve accuracy and efficiency of computed results
- Use of decoupling-based approach to preserve structure and semantics of MLNs
- Extensive experimental analysis on large number of synthetic and real-world-like graphs with diverse characteristics.
- Accuracy and Efficiency comparisons with ground truth and naive approach

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 introduces the decoupling approach for MLN analysis. Section 3.1 provides the challenges of decoupling-based approach for a global metric. Section 4 discusses the challenges in computing the closeness centrality nodes in MLNs. Section 5 describes the proposed heuristics for computing closeness centrality of an MLN. Section 6 describes the experimental setup and the data sets. Section 6.1 discusses result analysis followed by conclusions in Section 7.

2 RELATED WORK

Due to the rise in popularity and availability of complex and large real-world-like data sets, there is a critical need for modeling them as graphs and analyzing them in different ways. The centrality measure of MLN provides insight into different aspects of the network. Though there have been a plethora of studies in centrality detection for simple graphs, not many studies have been done on detecting central entities in multilayer networks. Existing studies conducted on detecting central entities in multilayer networks are *use-case specific* and no common framework exists which can be used to address the issue of detecting central entities in a multilayer network.

(Cohen et al., 2014) proposes an approach to find top k closeness centrality nodes in large graphs. This approximation-based approach has higher efficiency under certain circumstances. Even though the algorithm works on large graphs, unfortunately, it does not work in the case of multi-layer networks. In (De Domenico et al., 2015; Solé-Ribalta et al., 2016), authors capitalize on the tensor formalism, recently proposed to characterize and investigate complex topologies, to show how closeness centrality and a few other popular centrality measures can be extended to multiplexes. In (Cohen et al., 2014), authors propose a sampling-based approach to estimate the closeness centrality nodes in undirected and directed graphs with acceptable accuracy. The proposed approach takes linear time and space but it is a *main memory-based algorithm*. The authors of (Sariyüce et al., 2013) propose an incremental algorithm that can dynamically update the closeness centrality nodes of a graph in case of edge insertion and deletion. The algorithm has a lower memory footprint compared to traditional closeness centrality algorithms and provides massive speedup when tested on real-world-like graph data sets. In (Du et al., 2015), the authors

propose closeness centrality algorithms where they use effective distance instead of the conventional geographic distance and binary distance obtained by Dijkstra's shortest path algorithm. This approach works on directed, undirected, weighted, and unweighted graphs. In (Putman et al., 2019), authors compute the exact closeness centrality values of all nodes in dynamically evolving directed and weighted networks. This approach is parallelizable and achieves a speedup of up to 33 times.

Most of the methods to calculate closeness centrality are **main memory-based and not suitable for large graphs**. In (Santra et al., 2017b), the authors propose a decoupling-based approach where each layer can be analyzed independently and in parallel and calculate graph properties for a HoMLN using the information obtained for each layer. The proposed algorithms are based on the network decoupling approach which has been shown to be efficient, flexible, scalable as well as accurate.

3 DECOUPLING APPROACH FOR MLNs

Most of the algorithms available to analyze simple graphs for centrality, community, and substructure detection cannot be used for MLN analysis *directly*. There have been some studies that extend existing algorithms for centrality detection (e.g., page rank) to MLNs (De Domenico et al., 2015), but they try to **work on the MLN as a whole**. The network decoupling approach (Santra et al., 2017a; Santra et al., 2017b) used in this paper not only uses extant algorithms for simple graphs, but also uses a partitioning approach for efficiency, flexibility, and new algorithm development.

Briefly, existing approaches for multilayer network analysis convert or transform MLN into a single graph. This is done either by aggregating or projecting the network layers into a single graph. For homogeneous MLNs, edge aggregation is used to aggregate the network into a single graph. Although aggregation of an MLN into a single graph allows one to use extant algorithms (and there are many of them), due to aggregation, **structure and semantics of the MLN is not preserved resulting in information loss**.

The network decoupling approach is shown in Figure 2. It consists of identifying two functions: one for analysis (Ψ) and one for composition (Θ). Using the analysis function, each layer is analyzed independently (and in parallel). The results (which are termed partial from the MLN perspective) from each of the two layers are then combined using a composition function/algorithm to produce the results for the two layers of the HoMLN. This binary composition can be applied to MLNs with more than two layers. Independent analysis allows one to use existing algorithms on smaller graphs. The decoupling approach, moreover, adds efficiency, flexibility, and scalability.

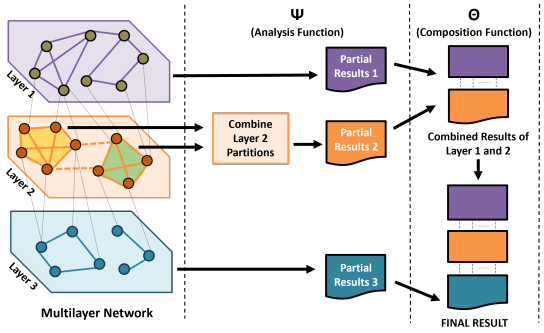


Figure 2: Overview of the network decoupling approach.

The network decoupling method preserves structure and semantics which is critical for drill-down and visualization of results. As each layer is analyzed independently, the analysis can be done in parallel reducing overall response time. Due to the MLN model, each layer (or graph) is likely to be smaller, requires less memory than the entire MLN, and provides clarity. The results of the analysis functions are saved and used for the composition. Each layer is analyzed only once. Typically, the composition function is less complex and is quite efficient, as shall be shown. Any of the existing simple graph centrality algorithms can be used for the analysis of individual layers. *Also, this approach is application-independent.*

When compared to similar single network approaches, achieving high accuracy with a decoupling approach is the challenge, especially for global measures. While analyzing one layer, identifying minimal additional information needed for improving accuracy due to composition is the main challenge. For many algorithms that have been investigated, there is a direct relation between the amount of additional information used and accuracy gained, however, this affects the efficiency.

3.1 Benefits and Challenges of Decoupling-Based Approach

For analyzing MLNs, currently, HoMLNs are converted into a single graph using aggregation approaches. Given two vertices u and v, the edges between them are aggregated into a single graph. The presence of an edge between vertices u and v depends on the aggregation function used. In Boolean AND

composed layers, if an edge is present between the same vertex pair u and v in *both* layers, then it will be present in the AND composed layer. Similarly, in Boolean OR composed layers, if an edge is present between the same vertex pair u and v in *at least one* of the layers in HoMLN, then the edge will be present in the OR composed layer.

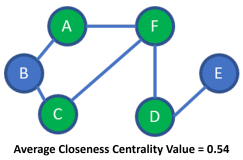
Both HoMLNs and HeMLNs are a set of layers of single graphs. Hence, the MLN model provides a natural partitioning of a large graph into layers of an MLN. The layer-wise analysis as the basis of the decoupling approach has several benefits. First, the entire network need not be loaded into memory, only a smaller layer. Second, the analysis of the individual layers can be parallelized decreasing the total response time of the algorithm. Finally, the computation used in the composition function (Θ) is based on intuition which is embedded into the heuristic and requires significantly less computation than Ψ.

When analyzing an MLN, the accuracy depends on the information being kept (in addition to the output) during the analysis of individual layers. In terms of centrality measures, the bare minimum information that can be kept from each layer is the high centrality (greater than average centrality value) nodes along with their centrality values. Retaining the minimal information, *local centrality measures* like degree centrality can be calculated relatively easily with high accuracy (Pavel et al., 2022) (Pavel. et al., 2022).

However, the calculation of a global measure, such as closeness centrality nodes, requires information of the entire MLN. *This compounds the difficulty of computation of closeness centrality of an MLN in the decoupling approach partial information used for estimation of the result will greatly impact the accuracy. Identification of useful minimal information and the intuition behind that are the primary challenges.*

4 CLOSENESS CENTRALITY: CHALLENGES

The closeness centrality value of a node v describes how far are the other nodes in the network from v or how fast or efficiently a node can spread information through the network. For example, when an internet service provider considers choosing a new geolocation for their servers, they might consider a city that is geographically closer to most cities in the region. An airline is interested in identifying a city for their hub that connects to other important cities with a minimum number of hops or layovers. For both, computing the closeness centrality of the network is the answer.



Node	Sum of *Distances or Shortest Path Lengths*	Normalized Closeness Centrality Value
A	9	0.56
B	11	0.45
C	9	0.56
D	9	0.56
E	13	0.38
F	7	0.71

Figure 3: Closeness centrality for a small toy graph.

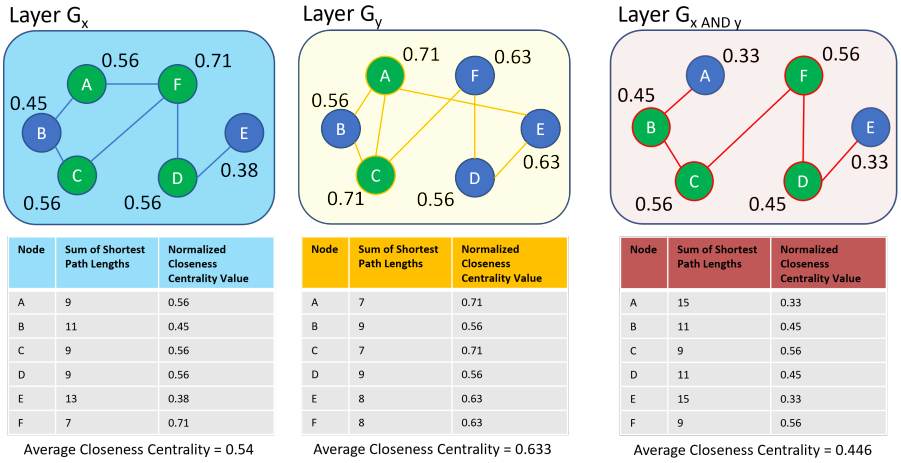
The closeness centrality score/value of a vertex u in a network is defined as,

$$CC(u) = \frac{n-1}{\sum_v d(u,v))} \quad (1)$$

where n is the total number of nodes and $d(u,v)$ is the shortest distance from node u to some other node v in the network. *The higher the closeness centrality score of a node, the more closely (distance-wise) that node is connected to every other node in the network.* Nodes with a closeness centrality score higher than the average are considered **closeness centrality nodes (or CC nodes)**. This definition of closeness centrality is only defined for graphs with a *single connected component*. The closeness centrality is defined for both directed and undirected graphs. In this paper, for an algorithm based on the decoupling approach, the focus is on the problem of finding high (same or above average) closeness centrality nodes of Boolean AND aggregated layers of an MLN for undirected graphs. Even though closeness centrality is not well defined for graphs with multiple connected components, the heuristics work for networks where each layer could consist of multiple connected components or the AND aggregated layer has multiple connected components. The proposed heuristics consider the normalized closeness centrality values over the connected component in the layers (Wasserman et al., 1994).

For closeness centrality discussed in this paper, the **ground truth (GT)** is calculated as follows: i) Two layers of the MLN are aggregated into a single graph using the Boolean AND operator and ii) Closeness centrality nodes of the aggregated graph are calculated using an existing algorithm. *The same algorithm is also used on the layers for calculating CC nodes of each layer.*

For finding the ground truth CC nodes and identifying the CC nodes in the layers, the NetworkX package (Hagberg et al., 2008) implementation of closeness centrality (Freeman, 1978) (Wasserman et al., 1994) is used. The implementation of the closeness centrality algorithm in this package uses breadth-first search (BFS) to find the distance from each vertex to every other vertex. For disconnected graphs, if a node is unreachable, a distance of 0 is assumed and finally, the obtained scores are normalized using Wasserman



Node	Sum of Shortest Path Lengths	Normalized Closeness Centrality Value
A	9	0.56
B	11	0.45
C	9	0.56
D	9	0.56
E	13	0.38
F	7	0.71

Average Closeness Centrality = 0.54

Node	Sum of Shortest Path Lengths	Normalized Closeness Centrality Value
A	7	0.71
B	9	0.56
C	7	0.71
D	9	0.56
E	8	0.63
F	8	0.63

Average Closeness Centrality = 0.633

Node	Sum of Shortest Path Lengths	Normalized Closeness Centrality Value
A	15	0.33
B	11	0.45
C	9	0.56
D	11	0.45
E	15	0.33
F	9	0.56

Average Closeness Centrality = 0.446

Figure 4: Layer G_x, G_y, and AND-aggregated layer created by G_x and G_y, G_{xANDy} (which is computed as G_x *AND* G_y) with the respective closeness centrality of each node. The nodes highlighted in green have above average closeness centrality values in their respective layers.

and Faust approximation which prioritizes the closeness centrality score of vertices in larger connected components (Wasserman et al., 1994). For a graph with V vertices and E edges, the time complexity of the algorithm is $O(V(V+E))$.

In addition to ground truth, the **naive approach** is used for comparing the accuracy of proposed heuristic-based approaches. In the naive approach, we take the intersection (as AND aggregation is being used for GT) of the CC nodes generated for each layer independently using the same algorithm. The result of the intersection of nodes is considered as the CC nodes for the MLN. The naive approach is the simplest form of composition (using the decoupling approach) and does not use any additional information other than the CC nodes from each layer. The hypothesis is that the naive approach is going to perform poorly when the topology of the two layers is very different. Observation 1 illustrates that the naive composition approach is **not guaranteed** to give ground truth accuracy, due to the *generation of false positives and false negatives.* One situation where naive accuracy coincides with the ground truth accuracy is when the two layers are identical. The naive accuracy will fluctuate with respect to ground truth without reaching it in general.

Observation 1. *A node that has above average closeness centrality value in the AND-aggregated layer created by G_x and G_y is not guaranteed to have above average closeness centrality value in one or both of the layers G_x and G_y.*

Example: For single connected component graphs, assume that an arbitrary node, say u, *does not* have above average centrality value in layer G_x and layer G_y. This means that node u has longer paths to reach every other node in the individual networks. That is, there exist other nodes, say v, which cover the entire *individual* networks through shorter paths, thus having high closeness centrality values. There can be scenarios in which these other nodes (v) *do not have enough short paths* that exist in *both* layer G_x and layer G_y, and as a result bring down their closeness centrality values (and also the average) in the aggregated layer, G_{xANDy}. Here, if the node u, has common paths from the two layers that are shorter than the common paths for other nodes, then its closeness centrality value will go above average in the AND aggregated layer.

This scenario is exemplified by node **B** in Figure 4, which in spite of not having above average closeness centrality value in layers G_x and G_y, has above average closeness centrality value in the AND-aggregated layer, G_{xANDy}. It can be observed that the closeness centrality value for other nodes decreased significantly as they did not have enough common shorter paths, however, node **B**, maintained enough common short paths, thus pushing its closeness centrality value above the average in the AND aggregated layer.

Hence, it illustrates that a node that has above average closeness centrality value in the AND-aggregated layer created by G_x and G_y is not guaranteed to have above average closeness centrality value in one or both of the layers G_x and G_y.

Lemma 1. *It is sufficient to maintain, for every pair of nodes, **all paths** from the layers, G_x and G_y, in order to find out the shortest path between the same nodes in the AND-aggregated layer created by G_x and G_y.*

Proof. Suppose, for every pair of nodes, say u and v, the set of *all* paths from the layers G_x and G_y, say $P_x(u,v)$ and $P_y(u,v)$, respectively, is being maintained. Then, by set intersection between $P_x(u,v)$ and $P_y(u,v)$, one can find out the shortest among the paths that exist between u and v in *both* the layers, G_x and G_y, which is basically the shortest path between them in the ANDed graph (G_{xANDy}) as well. The common path has to be part of the AND-aggregated graph by definition. If there are no common paths, the AND-aggregation is disconnected and the result still holds. Figure 4 shows an example. For the nodes **A** and **F**, if it is maintained $P_x(A,F)$ = {<(A,F)>, <(A,B), (B,C), (C,F)>} from layer G_x and $P_y(A,F)$ = {<(A,C), (C,F)>, <(A,C), (B,C), (C,F)>, <(A,E), (E,D), (D,F)>} from layer G_y, then one can obtain the path <(A,C), (B,C), (C,F)> as the shortest common path, which is the correct shortest path between these nodes in the ANDed layer, G_{xANDy}. The shortest path from G_x between A and F cannot appear in the ANDed graph. This proves the Lemma 1 that it is sufficient to maintain all paths between every pair of nodes to find out the shortest paths in the ANDed layer. □

Based on Lemma 1, in the composition step, for every pair of nodes, the path sets from two layers need to be intersected to find out the shortest common path. In doing so the closeness centrality value for each node in the ANDed graph can be calculated. Clearly, if $G_x(V,E_x)$ and $G_y(V,E_y)$ are two layers then the number of paths between any two nodes will be $(V-2)!$, in the worst case. Thus, the composition phase will have a complexity of $O((V-2)!(V-2)!)$, which is *exponentially higher* than the ground truth complexity $O(V(V+min(E_x,E_y)))$ and defeats the entire purpose of the decoupling approach. Thus, **the challenge is to identify the minimum amount of information to gain the highest possible accuracy over the naive approach by reducing the number of false positives and false negatives, without compromising on the efficiency.**

5 CLOSENESS CENTRALITY HEURISTICS

Two heuristic-based algorithms have been proposed for computing CC nodes for Boolean AND aggregated layers using the decoupled approach. The accuracy and performance of the algorithms have been tested against the ground truth. Extensive experiments have been performed on data sets with varying graph characteristics to show that the solutions work for any graph and have much better accuracy than the naive approach. Also, the efficiency of the algorithms is significantly better than ground truth computation. The solution can be extended to MLNs with any number of layers, where the analysis phase is applied once and the composition phase is applied as a function of pairs of partial layer results, iteratively.

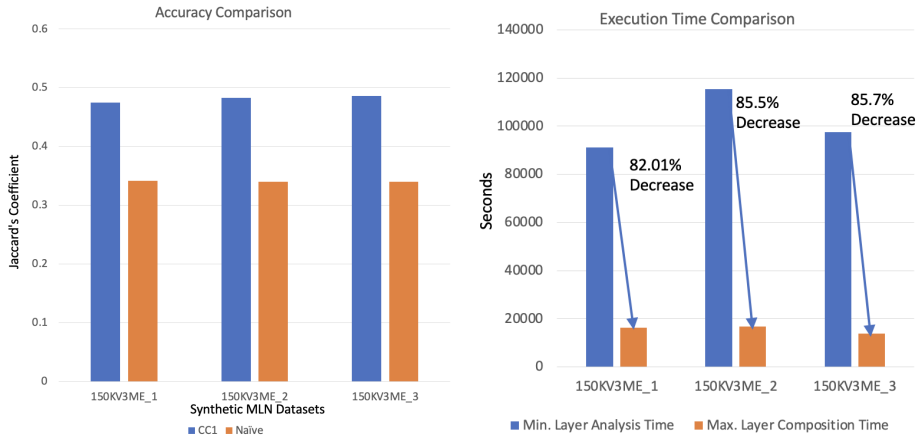
Jaccard coefficient is used as the measure to compare the accuracy of the solutions with the ground truth. Precision, recall, and F1-score have also been used as evaluation metrics to compare the accuracy of the solutions. For performance, the execution time of the solution is compared against that of the ground truth. The ground truth execution time is computed as: time required to aggregate the layers using AND composition function + time required to identify the CC nodes on the combined graph. The time required for the proposed algorithms using the decoupling approach is: **max**(layer 1, layer 2 analysis times) + composition time. **The efficiency results do not change much even if the max is not used.**

5.1 Closeness Centrality Heuristic CC1

Intuitively, CC nodes in single graphs have high degrees (*more paths go through it and likely more shortest paths*) or have neighbors with high degrees (similar reasoning.) In the ANDed graph (ground truth graph), CC nodes that are common among both layers have a high chance of becoming a CC node. Moreover, if a common CC node has high overlap of neighborhood nodes from both layers with above average degree and low average sum of shortest path (SP) distances, there is a high chance of that node becoming a CC node in the ANDed graph. Using this intuition and observation, heuristic CC1 has been proposed for identifying CC nodes for two layers as an MLN.

As discussed earlier, in the decoupling approach the analysis function Ψ is used to analyze the layers once and the partial results and additional information in the composition function Θ are used to obtain intermediate/final results. For `CC1`, after the analysis phase (Ψ) on each layer (say, G_x) for each node (say, u), its degree ($deg_x(u)$) and sum of shortest path distances ($sumDist_x(u)$), and the one-hop neighbors ($NBD_x(u)$) are maintained if u is a CC node, that is $u \in CH_x$. When calculating $sumDist_x(u)$, if a node v is unreachable (which can happen if the graph/layer has multiple disconnected components), the distance $dist(u,v) = n$ where n is the number of vertices in the layer (*same in each layer – HoMLN*) is considered. This is because the maximum possible path length in a graph with n nodes is (n-1).

In the composition phase (Θ), for each vertex



(a) Accuracy.: CC1 vs. Naive.

(b) Min Ψ vs. Max Θ Times.

Figure 5: Accuracy and Execution Time Comparison: CC1 vs. Naive.

u in each layer (say, G_x), the degree distance ratio ($degDistRatio_x(u)$) is calculated with respect to the second layer (say, G_y) to estimate the likelihood of a node to be a CC node in the ANDed graph (ground truth graph) as per equation 2.

$$degDistRatio_x(u) = \frac{sumDist_x(u)}{min(deg_x(u), deg_y(u))} \quad (2)$$

A smaller value of this ratio (i.e. smaller sum of distances and/or higher degree) means the vertex u has a higher chance of becoming a CC node in the ANDed graph. When calculating the $degDistRatio_x(u)$, the degree is estimated for vertex u in the ANDed graph, by taking into account the upper bound of degree, which is $min(deg_x(u), deg_y(u))$. Instead of using the degree of the nodes in each layer, using the estimated degree of the ground truth graph gives a better approximation of the ratio value of the sum of distance and degree for the ground truth graph. Due to *decrease* in the edges in the ground truth graph, the average sum of distances will increase (as on average, paths will get longer). As a result, the average degree distance ratio for the ANDed graph, G_{xANDy} is estimated using equation 3.

$$avgDegDistRatio_{xANDy} = max(avgDegDistRatio_x, avgDegDistRatio_y) \quad (3)$$

For each CC node in each layer, the set of *central* one-hop nodes, i.e. nodes having the *degDistRatio* less than the $avgDegDistRatio_{xANDy}$, is calculated. Finally, those *common* CC nodes from the two layers, which have a *significant overlap* among the *central* one-hop neighbors are identified as the CC nodes of the ANDed graph, which is given by the set CH'_{xANDy}. The complexity of the composition algorithm is dependent on the final step where the overlaps of CC nodes and their one-hop neighborhoods is considered. The algorithm will have the worst case complexity of $O(V^2)$ if both layers are complete graphs consisting of V vertices. Based on the wide variety of data sets used in experiments, it is believed that the composition algorithm will have an average case complexity of $O(V)$. We are not showing the CC1 composition algorithm due to space constraints and decided to include a better composition algorithm (CC2.)

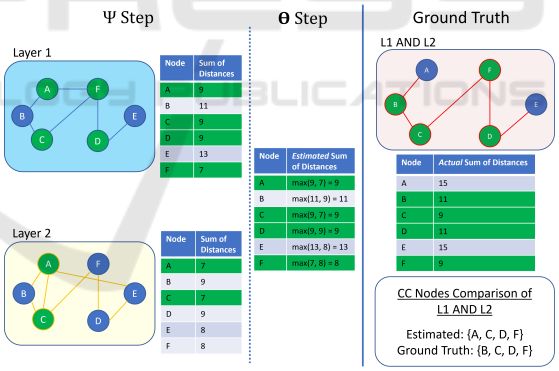


Figure 6: Intuition behind CC2 with example.

Discussion. Experimental results for CC1 are shown in Section 6.1. Figure 5 shows the accuracy (Jaccard coefficient) of CC1 compared against the naive approach on a subset of the synthetic data set 1 (details in Table 1). Although there is significant improvement in accuracy as compared to naive, it is still below 50%. This is seen as a drawback of the heuristic. Also, composition time is somewhat high for large graphs. In addition, keeping one-hop neighbors of the CC nodes of both layers is a significant amount of additional information. Figure 5b shows the *maximum composition time* against the *minimum analysis time* of the layers (worst case scenario). Even though

the composition time takes less time than the time it takes for the analysis of layers (one-time cost), the goal is to further *reduce the composition time and the additional information necessary for the composition* without making any major sacrifice to accuracy.

To overcome the above, *Closeness Centrality Heuristic* `CC2` *has been developed, which keeps less information than* `CC1`*, has faster composition time, and provides better (or same) accuracy.*

5.2 Closeness Centrality Heuristic CC2

For heuristic `CC1`, the estimated value of $avgDegDistRatio_{xANDy}$ is less than or equal to the actual average degree-distance ratio of the AND aggregated layer, so false negatives will be generated. The composition step of `CC1` is also computationally expensive and requires a lot of additional information. Furthermore, `CC1` cannot identify all CC nodes of the ground truth graph. Closeness centrality heuristic 2 or `CC2` has been designed to address these issues.

The design of `CC2` is based on estimating the sum of shortest path (SP) distances for vertices in the ground truth graph. If the sum of SP distances of a vertex in the individual layers is known, the upper and lower limit of the sum of SP distances for that vertex in the ground truth graph can be estimated. This idea can be intuitively verified. Let the sum of SP distances for vertex u in layer G_x (G_y) be $sumDist_x(u)$ ($sumDist_y(u)$.) Let the set of layer G_x (G_y) edges be E_x (E_y.) In the ground truth graph, the upper bound for the sum of SP distances of a vertex u is going to be ∞, if the vertex is disjoint in any of the layers. If $E_x \cap E_y = E_x$, the sum of SP distances for any vertex u in the ground truth graph is going to be $sumDist_x(u)$. If $E_x \subseteq E_y$, then the ground truth graph will have the same edges as layer x, and sum of SP distances for a vertex u in the ground truth graph will be same as the sum of SP distances for that vertex in layer G_x. When $E_x \cap E_y = E_x$, for any vertex u, $sumDist_x(u) \geq sumDist_y(u)$ because layer G_x will have less edges than layer G_y and average length of SPs in layer G_x will be higher than average length of SPs in layer G_y. Similarly, when $E_x \cap E_y = E_y$, the sum of SP distances for any vertex u in the ANDed graph is going to be $sumDist_y(u)$ and $sumDist_y(u) \geq sumDist_x(u)$. From the above discussion, it can be said that the sum of SP distances of a vertex u in the ANDed graph is between $max(sumDist_x(u), sumDist_y(u))$ and ∞.

Algorithm 1 shows the composition step for the heuristic `CC2`. In this composition function, it is assumed that the estimated sum of SP distances of vertex u is $estSumDist_{xANDy}(u)$. $sumDist_x(u)$ and $sumDist_y(u)$ are the sum of SP of node u in layer G_x and layer G_y respectively, and CH'_{xANDy} is the estimated set of CC nodes of the ANDed layer. Figure 6 illustrates how the composition function using CC2 is applied on a HoMLN with two layers. For each node u, the sum of SP is estimated, which is the maximum sum of SP of node among the two layers. In the example, node A has a sum of SP as 9 and 7 in layer 1 and layer 2, respectively. The estimated sum of SP of node A will be 9 in the ANDed layer. Similarly, the sum of SP of other vertices can be estimated. Once the estimated sum of SP of all the vertices in the ANDed graph is completed, Equation 1 can be used to calculate the CC values of the nodes in the ANDed layer. As the CC value for the ANDed layer is calculated using the estimated sum of SP, one can either take the CC nodes with above-average closeness centrality scores or take the top-k CC nodes. For an MLN with V nodes in each layer, the *worst-case complexity of the composition algorithm for* `CC2` *is* $O(V)$.

Algorithm 1: Procedure for Heuristic CC2.

INPUT: $deg_x(u)$, $deg_x(u)$, $sumDist_x(u)$, $sumDist_y(u)$ $\forall u$; CH_x, CH_y
OUTPUT: CH'_{xANDy}: estimated CC nodes in ANDed graph

1: **for** u **do**
2: $estSumDist_{xANDy}(u) \leftarrow max(sumDist_x(u), sumDist_y(u))$
3: **end for**
4: Calculate CH'_{xANDy} using $estSumDist_{xANDy}$

Discussion. Experimental results for `CC2` are shown in Section 6.1. In general, this heuristic gave better recall as compared to the naive and `CC1` approach by being able to decrease the number of false negatives (shown in Table 4). Moreover, in terms of accuracy, both Jaccard coefficient and F1-score (shown in Table 5) are better for `CC2` as compared to Naive and CC1 approach, (except a few special cases details on which are in section 6.1). Also it has been shown empirically that `CC2` is much more computationally efficient than `CC1`.

6 EXPERIMENTAL ANALYSIS

The implementation has been done in Python using the NetworkX (Hagberg et al., 2008) package. The experiments were run on SDSC Expanse (Towns et al., 2014) with single-node configuration, where each node has an AMD EPYC 7742 CPU with 128 cores and 256GB of memory running the CentOS Linux operating system.

For evaluating the proposed approaches, both synthetic and real-world-like data sets were used. Synthetic data sets were generated using PaRMAT (Kho-

Table 1: Summary of Synthetic Data Set-1.

Base Graph #V, #E	G_{ID}	Edge Dist. % *L1%,L2%*	#Edges		
			L1	L2	L1 AND L2
50KV, 250KE	1	70,30	224976	124988	50319
	2	60,40	149982	199983	50392
	3	50,50	174980	174977	50422
50KV, 500KE	4	70,30	399962	199986	51374
	5	60,40	249978	349954	51458
	6	50,50	299971	299960	51541
50KV, 1ME	7	70,30	349955	749892	55158
	8	60,40	649918	449935	55647
	9	50,50	549933	549922	55896
100KV, 500KE	10	70,30	249989	449970	100412
	11	60,40	299986	399978	100494
	12	50,50	349983	349981	100493
100KV, 1ME	13	70,30	799937	399978	101695
	14	60,40	699948	499969	101822
	15	50,50	599958	599964	101998
100KV, 2ME	16	70,30	699949	1499899	106389
	17	60,40	1299914	899926	107141
	18	50,50	1099924	1099923	107785
150KV, 750KE	19	70,30	674971	374979	150398
	20	60,40	449982	599970	150447
	21	50,50	524978	524975	150475
150KV, 1.5ME	22	70,30	1199942	599978	151684
	23	60,40	749968	1049954	151883
	24	50,50	899950	899956	152005
150KV, 3ME	25	70,30	1049951	2249888	156501
	26	60,40	1349920	1949906	157295
	27	50,50	1649922	1649909	157602

rasani et al., 2015), a parallel version of the popular graph generator RMAT (Chakrabarti et al., 2004) which uses the Recursive-Matrix-based graph generation technique.

For diverse experimentation, for each base graph, 3 sets of synthetic data sets have been generated using PaRMAT. The generated synthetic data set consists of 27 HoMLNs with 2 layers with varying edge distribution for the layers. The base graphs start with 50K vertices with 250K edges and go up to 150K vertices and 3 million edges[1]. In the first synthetic data set, one layer (L1) follows *power-law degree distribution* and the other one (L2) follows *normal degree distribution*. In the second synthetic data set, both HoMLN layers have *power-law degree distribution*. In the final synthetic data set, both layers have *normal degree distribution*. For each of these, 3 edge distributions percentages (70, 30; 60, 40; and 50, 50) are used for a total of **81 HoMLNs of varying edge distributions, number of nodes and edges** for experimentation.

Table 1 shows the details of the different 2-layer MLNs from the first synthetic data set (L1: power-law, L2: normal) used in the experiments. The other two synthetic data sets have similar node and edge distributions.

For the real-world-like data set, the network layers are generated from real-world like monographs using a random number generator. The real-world-like graphs are generated using RMAT with parameters to mimic real world graph data sets as discussed in (Chakrabarti, 2005). As a result, the graphs have multiple connected components and also their ground truth graph.

6.1 Result Analysis and Discussion

In this section, the results of the experiments have been presented. The two proposed heuristics have been tested on synthetic and real-world-like data sets with diverse characteristics. As a measure of accuracy, the Jaccard coefficient has been used. The precision, recall, and F1 scores for the proposed heuristics have also been compared. *As a performance measure, the time taken by the decoupling approach has been compared with the time taken to compute the ground truth (as defined earlier in Section 4.) In addition, the significance of the decoupling approach has also been highlighted by comparing the maximum composition time of the proposed algorithms with the minimum analysis time of the layers.* The accuracy of the algorithms is compared against the naive approach that serves as the baseline for comparison.

Table 2: Accuracy Improvement of `CC1` and `CC2` over Naive.

Data Set	Mean Accuracy			
	`CC1`	`CC2`	`CC1` vs. Naive	`CC2` vs. Naive
Synthetic-1	43.56%	46.77%	+52.57%	**+63.83%**
Synthetic-2	55.95%	55.20%	**+9.77%**	+8.30%
Synthetic-3	48.87%	50.90%	+47.55%	**+53.65%**
Real-world-like	88.71%	88.2%	+7.36%	+5.7%

Accuracy. Figure 7 illustrates the accuracy of both the heuristics and the naive approach against the ground truth for the synthetic data set-1. As one can see, the accuracy of the heuristics is better than the

[1]Graph sizes larger than this could not be run on a single node due to the number of hours allowed and other limitations of the XSEDE environment.

naive approach in all cases. In most cases, **CC2 performs better than CC1**. The accuracy of CC1 increases with the graph density. A similar trend has been observed in other synthetic data sets as well, where the **proposed CC1 and CC2 heuristics perform better than the naive approach**.

Figure 8 shows the accuracy of the algorithms on real-world-like data sets (distributions mimic real-world networks(Chakrabarti, 2005)). Across all data sets, both heuristics have **more than 80% accuracy**. The accuracy of the heuristics does not go below the naive approach even for *disconnected graphs*. This is significant, as the accuracy of the heuristics based on intuition is pretty good for real-world-like data sets. Although some of them show better accuracy for CC1 as compared to CC2, the efficiency improvement of CC2 is an order of magnitude better than CC1 (see Figure 9).

Table 2 shows the mean accuracy and average percentage gain in accuracy over the naive approach for the synthetic data sets and the real-world-like data set. For all synthetic data sets, the proposed heuristics *significantly* outperform the naive approach as shown. The least improvement in accuracy as compared to the naive approach is only when both layers have power-law degree distribution. Even for the real-world-like data set which has a high accuracy for the naive approach, the heuristics perform better than the naive approach.

Table 3: Precision of CC1 and CC2 over Naive.

Data Set	Mean Precision			
	CC1	CC2	CC1 vs. Naive	CC2 vs. Naive
Synthetic-1	60.98%	58.08%	**+1.52%**	-3.29%
Synthetic-2	74.39%	72.02%	-3.10%	-6.18%
Synthetic-3	51.99%	52.02%	+0.006%	**+0.0718%**

Precision. After comparing the precision values received using the proposed heuristics against the ones from the naive approach for synthetic dataset-1, it is observed that CC1 has overall better precision compared to the naive approach and CC2. In general, it can be observed from Table 3, that across different types of HoMLNs, CC1 and CC2 give high precision values, ranging from **51% to 74%**. However, the improvement over naive is *marginal* in most of the cases.

For synthetic data sets where one layer has normal degree distribution and the other layer has power-law degree distribution, CC2 precision drops slightly. CC2 was developed mainly to increase efficiency and preserve accuracy.

Recall. After comparing the recall values received using the proposed heuristics against the ones from the naive approach for synthetic data set 1, it is observed that CC2 has overall better recall compared to the naive approach and CC1. In general, it can be observed from Table 4 that across different types of HoMLNs, **both CC1 and CC2 have higher recall values than the naive approach**. Here, the heuristics achieve high recall values in the range from **49% to 73%**. It is observed that the proposed heuristics are able to decrease the false negatives more as compared to naive approach, which explains their marked improvement in terms of recall. However, the improvement over naive is *marginal* in most of the cases.

Table 4: Recall Improvement of CC1 and CC2 over Naive.

Data Set	Mean Recall			
	CC1	CC2	CC1 vs. Naive	CC2 vs. Naive
Synthetic-1	60.38%	71.01%	+70.87%	**100%**
Synthetic-2	72.79%	71.38%	**+16.71%**	+14.47%
Synthetic-3	48.87%	50.89%	+47.55%	**+53.65%**

Table 5: F1-Score Improvement of CC1 and CC2 over Naive.

Data Set	Mean F1-Score			
	CC1	CC2	CC1 vs. Naive	CC2 vs. Naive
Synthetic-1	60.58%	63.80%	+36.56%	**+43.80%**
Synthetic-2	71.67%	71.09%	**+6.21%**	+5.3%
Synthetic-3	50.22%	51.36%	+24.71%	**+27.53%**

F1-Score. On comparing the F1-score received using the proposed heuristics against the ones from the naive approach for synthetic data set 1, it is observed that CC2 has an overall better F1-score compared to the naive approach and CC1. It is established from Table 5 that across different types of HoMLNs, **both CC1 and CC2 have higher F1-scores than the naive approach**, reaching as high as **72%**.

Performance. The ground truth graph obtained from Boolean AND operation on layers of HoMLN will always have same or less number of edges than the individual layers as an edge will appear in the ground truth graph only if it is connected between the same nodes in both the layers. The NetworkX (Hagberg et al., 2008) package used here utilizes BFS to calculate the summation of distances from a node to every other node while calculating the normalized closeness centrality of the nodes. As the complexity of BFS depends on the number of vertices and edges in a graph, the ground truth will always require same or less time than the analysis time for the largest layer.

Although the sum of the analysis time of the layers may be more than that of the ground truth, one needs to only consider the maximum analysis time of layers as they can be done in parallel. Furthermore, the composition time is drastically less than the anal-

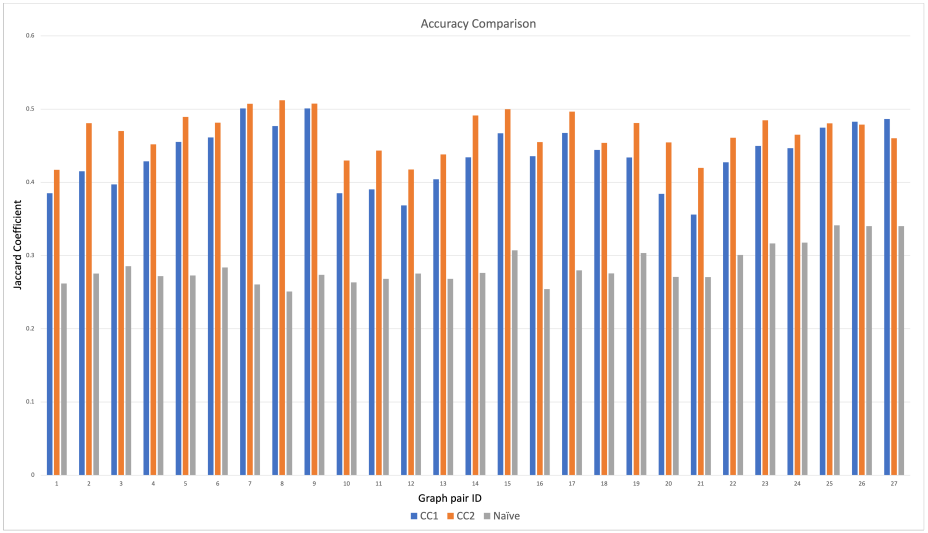


Figure 7: Accuracy Comparison for Synthetic Data Set-1 (Refer Table 1 where Graph Pair ID is G_{ID}).

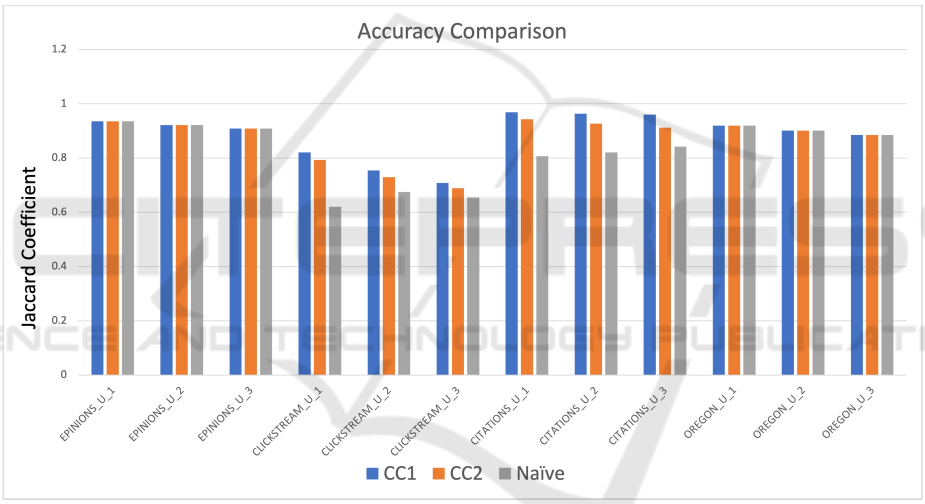


Figure 8: Accuracy of the heuristics `CC1` and `CC2` for the real-world-like data sets.

Figure 9: Performance Comparison of CC1 and CC2 on Largest Synthetic Data Set 1: Min. Ψ Time vs. Max. `CC1` Θ Time vs. Max `CC2` Θ Time (worst case scenario).

ysis time of any layer. Hence, *the minimum analysis time for layers is compared with the maximum composition time to show the worst case scenario*. As can be seen from Figure 9 *(plotted on log scale)*, the **maximum `CC1` composition time is at least 80% faster and `CC2` is an order of magnitude faster than the minimum analysis time!** In addition, the layer analysis is performed once and used for all subset CC node computation of n layers (which is exponential on n).

Discussion. Both proposed heuristics are better than the naive approach in terms of accuracy and way more efficient than ground truth computation. `CC2` is better than `CC1` if overall accuracy and efficiency are considered, but `CC1` performs better than `CC2` for high-density graphs and has better precision. The availability of multiple heuristics and their efficacy on accuracy and efficiency allows one to choose appropriate heuristics based on graph/layer characteristics.

7 CONCLUSIONS AND FUTURE WORK

In this paper, decoupling-based algorithms for *computing a global graph metric - closeness centrality - directly on MLNs* have been presented. Two heuristics (`CC1` and `CC2`) were developed to improve accuracy over the naive approach. `CC2` gives significantly higher accuracy than naive for graphs on a large number of synthetic graphs and graphs that are real-world-like with varying characteristics. `CC2` is extremely efficient as well. Future work is to extend the algorithms to more than 2 layers and improve accuracy.

ACKNOWLEDGMENTS

This work was partly supported by NSF Grants CCF-1955798 and CNS-2120393.

REFERENCES

(1993). Dblp: Digital bibliography & library project. dblp.org.

Boldi, P. and Vigna, S. (2014). Axioms for centrality. *Internet Mathematics*, 10(3-4):222–262.

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

Bródka, P., Skibicki, K., Kazienko, P., and Musiał, K. (2011). A degree centrality in multi-layered social network. In *2011 International Conf. CASoN*, pages 237–242.

Chakrabarti, D. (2005). *Tools for large graph mining*. Carnegie Mellon University.

Chakrabarti, D., Zhan, Y., and Faloutsos, C. (2004). R-mat: A recursive model for graph mining. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 442–446. SIAM.

Cohen, E., Delling, D., Pajor, T., and Werneck, R. F. (2014). Computing classic closeness centrality, at scale. In *Proceedings of the second ACM conference on Online social networks*, pages 37–50.

De Domenico, M., Solé-Ribalta, A., Omodei, E., Gómez, S., and Arenas, A. (2015). Ranking in interconnected multilayer networks reveals versatile nodes. *Nature Communications*, 6(1).

Du, Y., Gao, C., Chen, X., Hu, Y., Sadiq, R., and Deng, Y. (2015). A new closeness centrality measure via effective distance in complex networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(3):033112.

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239.

Hagberg, A., Swart, P., and S Chult, D. (2008). Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Halu, A., Mondragón, R. J., Panzarasa, P., and Bianconi, G. (2013). Multiplex pagerank. *PLOS ONE*, 8(10):1–10.

Khorasani, F., Gupta, R., and Bhuyan, L. N. (2015). Scalable simd-efficient graph processing on gpus. In *Proceedings of the 24th International Conference on Parallel Architectures and Compilation Techniques*, PACT '15, pages 39–50.

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., and Porter, M. A. (2014). Multilayer networks. *Journal of Complex Networks*, 2(3):203–271.

Pavel, H., Roy, A., Santra, A., and Chakravarthy, S. (2022). Degree centrality definition, and its computation for homogeneous multilayer networks using heuristics-based algorithms. In *International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management*, pages 28–52. Springer.

Pavel., H. R., Santra., A., and Chakravarthy., S. (2022). Degree centrality algorithms for homogeneous multilayer networks. In *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2022) - KDIR*, pages 51–62. INSTICC, SciTePress.

Pedroche, F., Romance, M., and Criado, R. (2016). A biplex approach to pagerank centrality: From classic to multiplex networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(6):065301.

Putman, K. L., Boekhout, H. D., and Takes, F. W. (2019). Fast incremental computation of harmonic closeness centrality in directed weighted networks. In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1018–1025.

Santra, A., Bhowmick, S., and Chakravarthy, S. (2017a). Efficient community re-creation in multilayer networks using boolean operations. In *International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland*, pages 58–67.

Santra, A., Bhowmick, S., and Chakravarthy, S. (2017b). Hubify: Efficient estimation of central entities across multiplex layer compositions. In *IEEE International Conference on Data Mining Workshops*.

Sariyüce, A. E., Kaya, K., Saule, E., and Çatalyiirek, Ü. V. (2013). Incremental algorithms for closeness centrality. In *2013 IEEE International Conference on Big Data*, pages 487–492.

Shi, Z. and Zhang, B. (2011). Fast network centrality analysis using gpus. *BMC Bioinformatics*, 12(1).

Solá, L., Romance, M., Criado, R., Flores, J., García del Amo, A., and Boccaletti, S. (2013). Eigenvector centrality of nodes in multiplex networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 23(3):033131.

Solé-Ribalta, A., De Domenico, M., Gómez, S., and Arenas, A. (2016). Random walk centrality in interconnected multilayer networks. *Physica D: Nonlinear*

Phenomena, 323-324:73–79. Nonlinear Dynamics on Interconnected Networks.

Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Lifka, D., Peterson, G. D., Roskies, R., Scott, J., and Wilkins-Diehr, N. (2014). Xsede: Accelerating scientific discovery. *Computing in Science and Engineering*, 16(05):62–74.

Wasserman, S., Faust, K., et al. (1994). Social network analysis: Methods and applications.