Decomposing graphs to balance virus spreading and efficiency using Graph Neural Networks

Magdalena Proszewska[1] and Michal Bujak, Marek Smieja, Jacek Tabor, Rafal Kucharski*[2]

[1]School of Informatics, University of Edinburgh, United Kingdom
[2]Faculty of Mathematics and Computer Science, Jagiellonian University, Poland

Short summary

Maintaining a high system performance while imposing virus preventing measures is a challenging task. We introduce a Deep Epidemic Efficeincy Network (DEEN) which balances the two opposing goals via graph partition. Our model optimises graph efficiency while meeting increasing levels of the epidemic threshold. We introduced our method to the ride-pooling service in New York City. By dividing 150 New York taxi travellers into four groups, our method increases the epidemic threshold by more than twofold at the cost of reducing utility only by 13%. We validate our model against other real-world examples: cross-region economic exchange in Poland and information sharing in a peer-to-peer network.
Keywords: graph neural network, graph partition, ride-pooling,

1 Introduction

The recent COVID-19 outbreak had a huge impact on our lives. Required social distancing impeded trade, production, shared transport capabilities. The situation is naturally represented on a graph where nodes portray people or economic centres while edges a contact between them. Studies for ant colonies and classes partitioning were conducted by Chen et al. (2019) and Kaiser et al. (2021), respectively.

With the increased network connectivity, the utility increases while the epidemic threshold decreases. Wu & Liu (2008) analysed the impact of the topological properties on spreading. We can measure the original network utility (the full potential when no constraints are applied) and utility in the decomposed network (certain edges are removed to separate the population into pairwise disjoint components). The challenge addressed by this study is to find the golden mean where the network maintains close to its original utility while imposing a high epidemic threshold. Our model is based on Graph Neural Networks introduced by Scarselli et al. (2008) and later further developed by Shuman et al. (2013), Kipf & Welling (2017) and Bianchi et al. (2020).

Our inspiring example was ride-pooling. As described in seminar papers by Santi et al. (2014) and Alonso-Mora et al. (2017), first we must find feasible combinations of travellers, which we later subset for the optimal matching. Our model decomposes the shareability graph (feasible combinations) without knowledge of how matching is done. It successfully partitions the graph into balanced components, achieving both a high epidemic threshold and close to the original performance. We validated the algorithm on two other examples: economic exchange between regions in Poland and a peer-to-peer computer network, where it delivered similar results.

2 Methodology

We denote $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as a weighted directed graph representing the system, $\mathbf{A} = (\mathbf{A}_{ij}) \in \mathrm{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ its adjacency matrix with weights. We define $\Delta = (\Delta_{ij}) \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{V}|}$ as

$$\Delta_{ij} = \begin{cases} 1, & \mathbf{A}_{ij} > 0 \\ 0, & \mathbf{A}_{ij} \leq 0. \end{cases} \tag{1}$$

Decomposition of a graph $\mathcal{G}$ can be defined as a subgraph $\mathcal{H} = (\mathcal{V}_H, \mathcal{E}_H)$, such that $\mathcal{V}_H = \mathcal{V}$, $\mathcal{E}_H \subseteq \mathcal{E}$ and $\mathcal{H}$ consists of disjointed subgraphs $\mathcal{H}_1, \mathcal{H}_2, ..., \mathcal{H}_k$, meaning $\bigcup \mathcal{V}_{\mathcal{H}_i} = \mathcal{V}_{\mathcal{H}}, \bigcup \mathcal{E}_{\mathcal{H}_i} = \mathcal{E}_{\mathcal{H}}$ and $\mathcal{V}_{\mathcal{H}_i} \cap \mathcal{V}_{\mathcal{H}_j} = \emptyset$, for $i \neq j$. We denote set of decompositions of graph $\mathcal{G}$ as $\mathcal{D}(\mathcal{G})$.

The effectiveness of the system represented by the graph is measured as the utility. It is a non-decreasing function ($\mathcal{H} \subseteq \mathcal{G} \implies U(\mathcal{H}) < U(\mathcal{G})$). The function can be not analytical and black-box (calculated with an external algorithm). For the pooling service it is vehicle kilometre reduction, for economic exchange a complex accessibility formula, for peer-to-peer network as a share of preserved connections.
We measure the virus spreading via epidemic threshold. We use the Susceptible-Infected-Susceptible epidemic model (Shi et al., 2008) and apply heterogeneous mean-field approach (Wang et al., 2017). For a connected graph $\mathcal{G}$ we define it as

$$ET(\mathcal{G}) = \frac{\sum_{v\in\mathcal{V}} \deg(v)}{\sum_{v\in\mathcal{V}} \deg(v)^2}. \tag{2}$$

Our method is grounded in decomposition, hence we generalise the epidemic threshold for a disconnected graph. Let $C(\mathcal{G}) = \{\mathcal{G}_i\}_i$ denote set of its connected, pair-wise disjoint components. We generalise $ET(\mathcal{G})$ as

$$ET(\mathcal{G}) = \sum_{\mathcal{G}_i\in C(\mathcal{G})} \frac{|\mathcal{V}_i|}{|\mathcal{V}|} ET(\mathcal{G}_i). \tag{3}$$

Optimisation problem

Let U be an (unknown) utility function on the weighted graph $\mathcal{G}$. Given a target epidemic threshold β, we seek a decomposition $\mathcal{H} \in \mathcal{D}(\mathcal{G})$, which maximises the utility U and has an epidemic threshold $ET(\mathcal{G})$ equal to or greater than β. Finding such decomposition $\mathcal{H}$ (clustering) guarantees that virus transmission will be reduced while the effectiveness of the system network will be maximal. This optimisation problem can be written as

$$\begin{aligned} \max_{\mathcal{H}\in\mathcal{D}(\mathcal{G})} \quad & U(\mathcal{H}), \\ \text{s.t.} \quad & ET(\mathcal{H}) \geq \beta. \end{aligned} \tag{4}$$

Deep Epidemic Efficiency Network

We introduce the following framework to solve the optimisation problem (6). To account for all our goals, we define the loss function comprising three elements: utility loss, virus spreading and regularisation factor (ensure balanced assignments). All three components are described later.
The model is built on Graph Convolutional Neural Networks (GCNN) (Kipf & Welling, 2017) with the output layer defined by the softmax function. To include feature information about a node in the propagation process, we modify the weight matrix by adding self-loops, i.e.

$$\widehat{\mathbf{A}} = \mathbf{A} + \delta\mathbf{I}, \tag{5}$$

where $\delta \in \mathbb{R}_+$ (Lampert & Scholtes, 2023). The GCNN returns the assignment matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{V}|\times K}$:

$$\mathbf{S} = \text{softmax}(\text{GCNN}(\widehat{\mathbf{A}})), \tag{6}$$

where $K \in \mathbb{N}_+$ is the resulting number of clusters. Due to the use of weighted edges, we do not experience training instability and over-smoothing, therefore, unlike Kipf & Welling (2017), we do not apply Laplacian normalisation.
We build a model for an arbitrary form of the utility function and make use of edge weights to approximate edge importance in the overall utility. Therefore, the first component of our loss is the utility loss expressed as

$$\mathcal{L}_u(\mathbf{S};\mathbf{A}) = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \sum_{j=1}^{|\mathcal{V}|} a_{ij}(1 - s_i s_j^T), \tag{7}$$

where $s_i \in \mathbb{R}^K$ is the i-th row of $\mathbf{S}$ representing cluster assignment of i-th node. This loss forces cluster assignments of nodes connected with high weight links closer to each other. We intentionally use a non-specific formula for the utility to ensure that our model is general, the function is both easily calculated and differentiable. Maximising weighted similarity is highly correlated with many typical examples of utility functions.

For a given assignment to clusters, we approximate the node degree as (note that specific edge weights are not relevant here, only their sign)

$$d_i = \sum_{j=1}^{|\mathcal{V}|} \sum_{k=1}^{K} \Delta_{ij} \cdot s_{ik} \cdot s_{jk} \tag{8}$$

and represent degree vector for graph $\mathcal{G}$ as

$$\mathbf{d} = \text{diag}(\Delta^T \mathbf{S}\mathbf{S}^T). \tag{9}$$

Then, we define the virus spreading loss for a connected graph as

$$\mathcal{L}_{vs}(\mathbf{S}; \mathbf{A}) = -\frac{\|\mathbf{d} + \mathbf{e}\|_1}{\|\mathbf{d} + \mathbf{e}\|_2^2}. \tag{10}$$

The additional 1 is added to each vector degree (self-loop with weight 1) to avoid $\mathcal{L}_{vs} \xrightarrow{d \to 0} -\infty$. However, since we operate on graphs that are not necessarily connected, we define virus spreading loss as a weighted sum:

$$\mathcal{L}_{vs}(\mathbf{S}; \mathbf{A}) = \sum_{(\mathcal{V}_i, \mathcal{E}_i) \in C(\mathcal{G})} \frac{|\mathcal{V}_i|}{|\mathcal{V}|} \mathcal{L}_{vs}(\mathbf{S}^{(i)}; \mathbf{A}^{(i)}), \tag{11}$$

where $C(\mathcal{G})$ denotes set of connected components of graph $\mathcal{G}$ and $\mathcal{V}_i$, $\mathcal{E}_i$, $\mathbf{S}^{(i)}$, $\mathbf{A}^{(i)}$ represent a set of nodes, a set of edges, a cluster assignment matrix and an adjacency matrix for component i, respectively.

Lastly, we use of collapse regularisation proposed by Tsitsulin et al. (2023) to prevent the trivial decomposition while not dominating the optimisation of the main objective. It is defined as

$$\mathcal{R}_c(\mathbf{S}) = \frac{\sqrt{K}}{|\mathcal{V}|} \left\| \sum_i \mathbf{S}_i^\top \right\|_F - 1, \tag{12}$$

where $\|\cdot\|_F$ denotes Frobenius norm. Without the collapse regularisation, clustering for our objectives has local minima (creating empty clusters) that trap the gradient-based optimisation.

Taking the above three components together, we arrive at the final expression of our loss function:

$$\mathcal{L}_{DEEN}(\mathbf{S}; \mathbf{A}) = \mathcal{L}_u(\mathbf{S}; \mathbf{A}) + \lambda \mathcal{L}_{vs}(\mathbf{S}; \mathbf{A}) + \mathcal{R}_c(\mathbf{S}), \tag{13}$$

where $\lambda \in \mathbb{R}_+$ balances virus spread and connectivity of a graph. We set $\lambda = 0.4$ for all of our experiments to reach the balance between virus spreading and effectiveness that we wanted. Higher λ pushes model to achieve higher epidemic threshold and lower effectiveness, while lower λ has an opposite effect. To uncover an optimal number of clusters to solve problem 4, we perform a binary search and find the minimal number such that it exceeds given epidemic threshold. For each case, we train the model with the same architecture: 3 non-normalised graph convolutional layers and 1 dense layer. We apply the ReLU activation function, Adam optimiser (with a learning rate of 0.001) and train till convergence (2000 epochs). To ensure that there are no degenerate solutions, we impose the maximum number of clusters to $\frac{n}{2}$ in ride-pooling experiment and 32 for the other two.

We compare our results to other partitioning algorithms: greedy algorithms by Clauset-Newman-Moore (Clauset et al., 2004) and Girvan-Newman (Girvan & Newman, 2002); spectral clustering; and other deep networks: MinCutPool (Bianchi et al., 2020), Just balance GNN (Bianchi, 2023) and Deep Modularity Network (DMoN) (Tsitsulin et al., 2023).

3 Results and discussion

For ride-pooling experiments, we use publicly available dataset of trip requests for Manhattan, New York[1]. We conduct the evaluation using 21 batches (8-8:30AM, 4-4:30PM and 12-12:30AM) from seven consecutive days, ranging from 78 to 204 trip requests. Using ExMAS algorithm (Kucharski & Cats (2020)), we obtain first the shareability graphs (feasible combinations of travellers) and

[1]Dataset available at `https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page`.

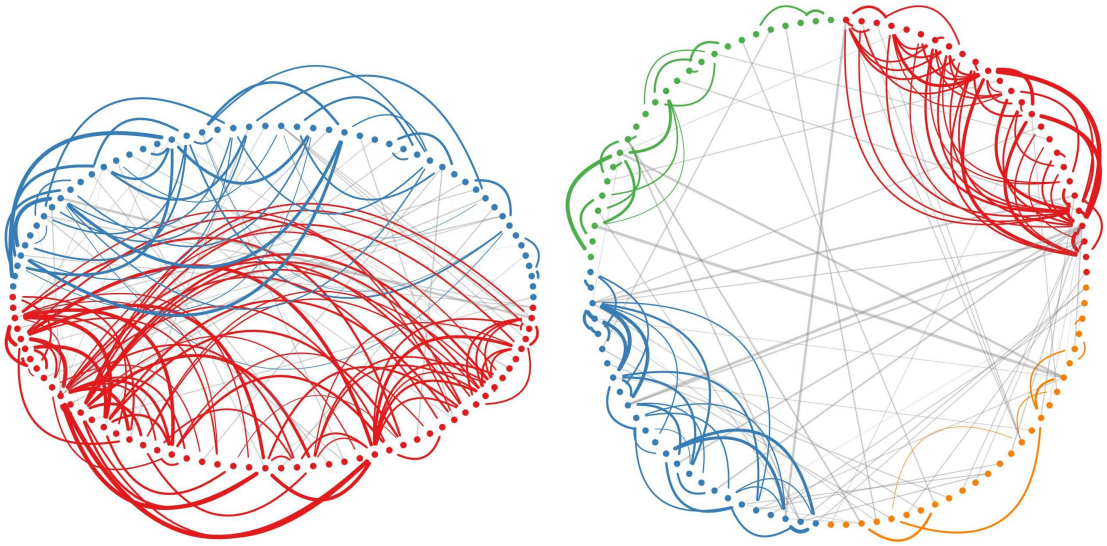
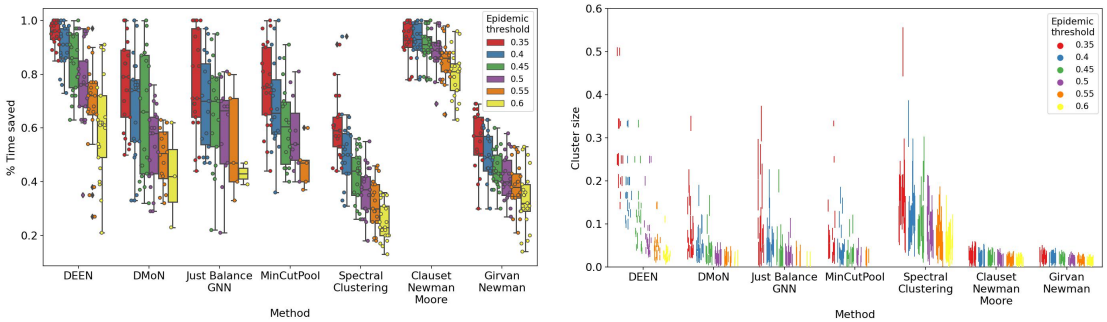


Figure 1: Method showcase: The network of 150 travellers in NYC using ride-pooling services (shared taxi offered by, e.g., UberPool). The nodes are travellers, linked if they can efficiently travel together. Depending on the desired epidemic threshold, DEENremoves the links marked gray and divides the network into two (left) or four (right) clusters, which increases the initial epidemic threshold of 0.13 to 0.35 and 0.47 at the expense of 0.03 and 0.13 of the original saved vehicle distance, respectively. Colours denote resulting clusters. Inner edges correspond to the potential travellers pairings that are not physically realised while the outer edges – a subset of the potential pairings that constitutes the optimal solution for ride-pooling.

then measurement of the maximum vehicle kilometre reduction. We decompose shareability graphs so that travellers assigned to different clusters can no longer share a trip. For the new, smaller set of feasible combinations, we perform matching (for details refer to ExMAS description) and find the optimal solution. Then we calculate the new performance value (saved vehicle kilometres).
For shareability graphs, we define edge weights (used in utility loss (7)) as

$$w(v,u) = \frac{t(v) + t(u) - t(v,u)}{t(v) + t(u)}, \tag{14}$$

where $t(v)$ and $t(u)$ denote individual travel time for passengers v and u, respectively (without ride-pooling) and $t(v,u)$ represents travel time when passengers share a ride. Note that $w(v,u)$ can be negative, when taking two passengers increases total travel time.



(a) Utility of graph decomposition measured in share of the original network utility preserved, with different virus spreading objectives (colours).

(b) Relative cluster sizes represented as lines. Middle point of each line represents average size of a cluster (the larger the better). Length of a line shows standard deviation of sizes (the smaller the better).

Figure 2: Decomposition results for ride-pooling shareability graphs.

Figure 2a presents what share of the original utility (travel time savings) was preserved after decomposition needed to achieve given epidemic threshold. The best results are the closest to 1, i.e. preserving nearly completely the original utility. Figure 2b shows relative sizes of clusters with

Epidemic threshold	0.3	0.4	0.5
DEEN	0.37	0.36	0.28
DmoN	-	-	-
Just Balance GNN	0.61	-	-
MinCutPool	-	-	-
Clauset-Newman-Moore	0.31	0.29	0.26

Table 1: Share of original utility preserved after decomposition of graph of Polish regions to reach respective epidemic threshold.

their deviation. Higher average size means that effectively fewer clusters were required to achieve a certain threshold. Shorter lines indicate all clusters are of a similar size (less degenerated). In the ride-pooling, if each cluster is to be served separately, it is particularly appreciated if their size is similar.
To properly evaluate model performance, we look at both figures. We note we decompose graphs of potential pairings of travellers, from which certain combinations effectively share rides. Greedy algorithms split graphs into many small subgraphs. They successfully sustain utility while imposing a high epidemic threshold. However, the large number of clusters indicates they uncover the most effective rides and put them in single clusters. The solution yields little application if the service provider aims to arbitrarily cluster potential clients (only a portion of potential clients will eventually use the service). Spectral clustering techniques create balanced subgraphs, however they do not preserve utility. Deep Networks balance the number of clusters (greater size) and the utility better, but overall DEEN achieves best results in terms of both utility and cluster sizes. Concluding, we recognise our method best as it offers the least numbers of clusters with small deviation and preserves the highest utility.

Other applications

We validate our method with other real-life examples. During the pandemic outbreak, people were forced to limit their social contacts and often limit their presence outside homes to absolute necessities. We study regional lockdown in Poland. Our goal is to maximise the potential economic exchange while minimising cross-region transmission. In the graph representation of the problem, nodes are regions and connections indicate immediate neighbourhood. The edge weight is defined as follows:

$$w(v,u) = \left(\frac{p(v)}{2 \cdot \max_{w \in \mathcal{N}(u)} p(w)} + \frac{p(u)}{2 \cdot \max_{w \in \mathcal{N}(v)} p(w)} \right). \tag{15}$$

The maximum term normalises the edge weight over its neighbours and 2 accounts for counting both ends.
To properly evaluate economic exchange, we apply accessibility formulas (following Levinson (1998)), which not only consider neighbours but also other reachable nodes:

$$U(v, \mathcal{R}_v) = \sum_{u \in \mathcal{R}_v / \{v\}} \frac{p(v) \cdot p(u)}{d(u,v)}, \tag{16}$$

where $p(u)$ is the population, $d(u,v)$ denotes distance between centres of regions and $\mathcal{R}_v$ denotes set of regions reachable from region v. Note that although initially, every region is accessible from any node v, this is not true after graph decomposition. Network's utility $\sum_v U(v, \mathcal{R}_v)$ will be reduced as our model decomposes regions into the clusters.
Girvan-Newman and spectral clustering baselines were not possible to evaluate due to the number of nodes and our limited resources. Table 1 shows that only DEEN and Clauset-Newman-More baseline were able to achieve high epidemic threshold (for maximum number of clusters set to 32 for Deep Network methods).
Moreover, Figure 3 highlights that DEEN creates components that are better balanced in terms of size. In particular, Clauset-Newman-Moore algorithm creates small set of large dominant components and a lot of small ones.
Examples of decompositions made by DEEN are shown in Figure 4.
We validate our method against the peer-to-peer Gnutella network Ripeanu et al. (2002). Edges represent connection between hosts. Our utility measure is a fraction of preserved connections. Quantitative results are presented in 2.

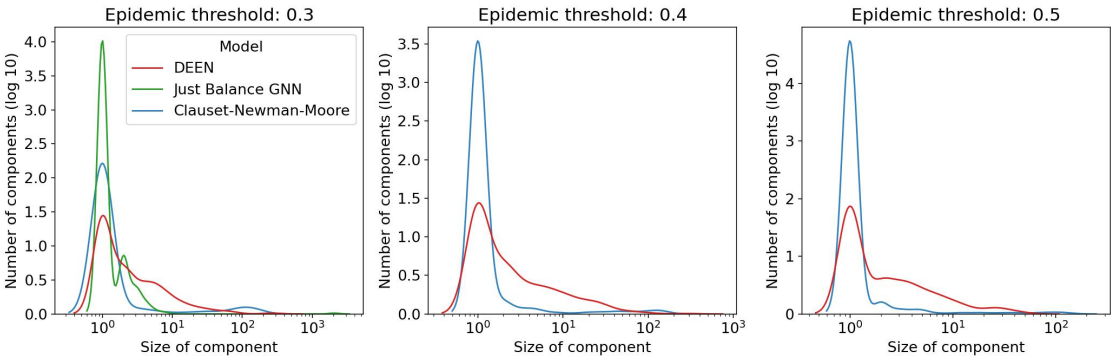
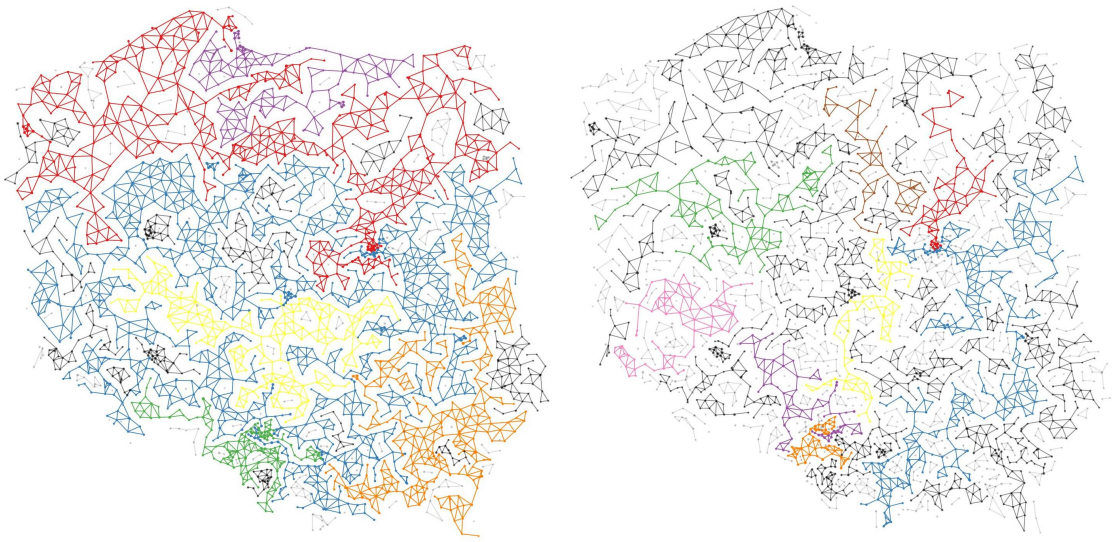


Figure 3: KDE of sizes of components for regional decomposition.



(a) (b)

Figure 4: Decompositions of Polish regions using DEEN. They increase epidemic threshold from 0.2 (original) to 0.27 (4a) and 0.37 (4b), and decreases utility to 52% and 36%, respectively. Components with at least 50 nodes are coloured, components with 10-49 nodes are grey and components with less than 10 nodes are faded.

	Utility			Clusters size (AVG ± SD)		
Epidemic threshold	0.4	0.5	0.6	0.4	0.5	0.6
DEEN	0.38	0.28	0.22	0.25 ± 0.01	0.17 ± 0.01	0.14 ± 0.01
DMoN	0.53	0.43	–	0.36 ± 0.09	0.29 ± 0.08	–
Just Balance GNN	0.59	0.39	–	0.42 ± 0.02	0.28 ± 0.06	–
MinCutPool	0.76	–	–	0.47 ± 0.13	–	–
Clauset-Newman-Moore	0.33	0.26	0.22	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00

Table 2: Average preserved utility after decomposition of P2P networks and size of obtained clusters described by average size and standard deviation (% of nodes). Note that 0.00 ± 0.00 means that the average size of cluster and the standard deviation was smaller than 1% of nodes. Reported numbers are averages over 9 P2P networks.

4 Conclusions

The Deep Epidemic Efficiency Network we introduced effectively decomposes ride-pooling shareability graphs. We achieve close to the original saved vehicle mileage (87%) while increasing the epidemic threshold by 240%. Our model provides an efficient way to limit contacts in shared mobil-

ity, such that the service performance is hardly hindered and the safety measures are significantly increased. DEEN is flexible and successfully tackled other real-life problems: maintaining economic exchange under lockdown and preventing the spread of computer virus. The model proves to be applicable to graphs of different sizes and topologies. Its tuneable parameters offer the user to amplify the role of either preserved utility or maximised safety.

Acknowledgements

This research was funded by National Science Centre in Poland program OPUS 19 (Grant Number 2020/37/B/HS4/01847)

References

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., & Rus, D. (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, *114*(3), 462–467.

Bianchi, F. M. (2023, jan). Simplifying clustering with graph neural networks. *Proceedings of the Northern Lights Deep Learning Workshop*, *4*. doi: 10.7557/18.6790

Bianchi, F. M., Grattarola, D., & Alippi, C. (2020). *Spectral clustering with graph neural networks for graph pooling.*

Chen, W.-N., Tan, D.-Z., Yang, Q., Gu, T., & Zhang, J. (2019). Ant colony optimization for the control of pollutant spreading on social networks. *IEEE Transactions on Cybernetics*, *50*(9), 4053–4065.

Clauset, A., Newman, M. E., & Moore, C. (2004). Finding community structure in very large networks. *Physical review E*, *70*(6), 066111.

Girvan, M., & Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, *99*(12), 7821–7826.

Kaiser, A. K., Kretschmer, D., & Leszczensky, L. (2021). Social network-based cohorting to reduce the spread of sars-cov-2 in secondary schools: a simulation study in classrooms of four european countries. *The Lancet Regional Health–Europe*, *8*.

Kipf, T. N., & Welling, M. (2017). *Semi-supervised classification with graph convolutional networks.*

Kucharski, R., & Cats, O. (2020). Exact matching of attractive shared rides (exmas) for system-wide strategic evaluations. *Transportation Research Part B: Methodological*, *139*, 285–310.

Lampert, M., & Scholtes, I. (2023). *The self-loop paradox: Investigating the impact of self-loops on graph neural networks.*

Levinson, D. M. (1998). Accessibility and the journey to work. *Journal of transport geography*, *6*(1), 11–21.

Ripeanu, M., Foster, I., & Iamnitchi, A. (2002). *Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design.*

Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S. H., & Ratti, C. (2014). Quantifying the benefits of vehicle pooling with shareability networks. *Proceedings of the National Academy of Sciences*, *111*(37), 13290–13294.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. *IEEE transactions on neural networks*, *20*(1), 61–80.

Shi, H., Duan, Z., & Chen, G. (2008). An sis model with infective medium on complex networks. *Physica A: Statistical Mechanics and its Applications*, *387*(8-9), 2133–2144.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., & Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, *30*(3), 83–98.

Tsitsulin, A., Palowitch, J., Perozzi, B., & Müller, E. (2023). *Graph clustering with graph neural networks.*

Wang, W., Tang, M., Stanley, H. E., & Braunstein, L. A. (2017). Unification of theoretical approaches for epidemic spreading on complex networks. *Reports on progress in physics*, *80*(3), 036603.

Wu, X., & Liu, Z. (2008). How community structure influences epidemic spread in social networks. *Physica A: Statistical Mechanics and its Applications*, *387*(2-3), 623–630.