Whole graph embedding: robustness and vulnerability

M. Manzo[1] M. Giordano[2] L. Maddalena[2] M. R. Guarracino[3,4,2]

[1]University of Naples "L'Orientale", Italy

[2]National Research Council, Italy

[3]University of Cassino and Southern Lazio, Italy

[4]National Research University Higher School of Economics, Russia

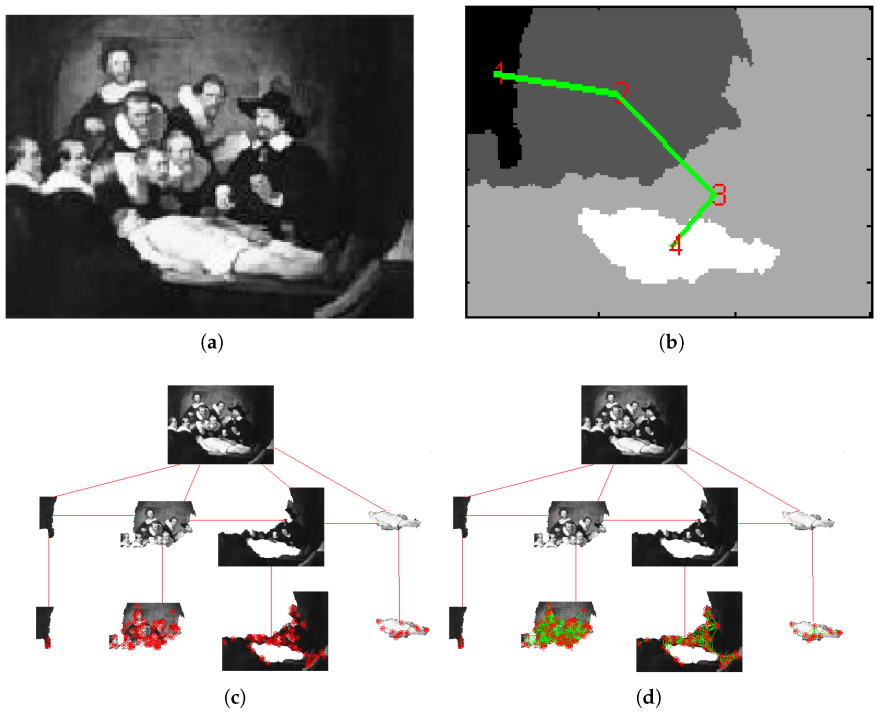
INdAM Workshop
How can Scientific Computing help to study Life Sciences?
13 September 2021

Roadmap

- Preliminaries.
- Adversarial machine learning.
- Graph adversarial attacks.
- Experiments.
- Conclusions and Future Works.

Graph-based data representation

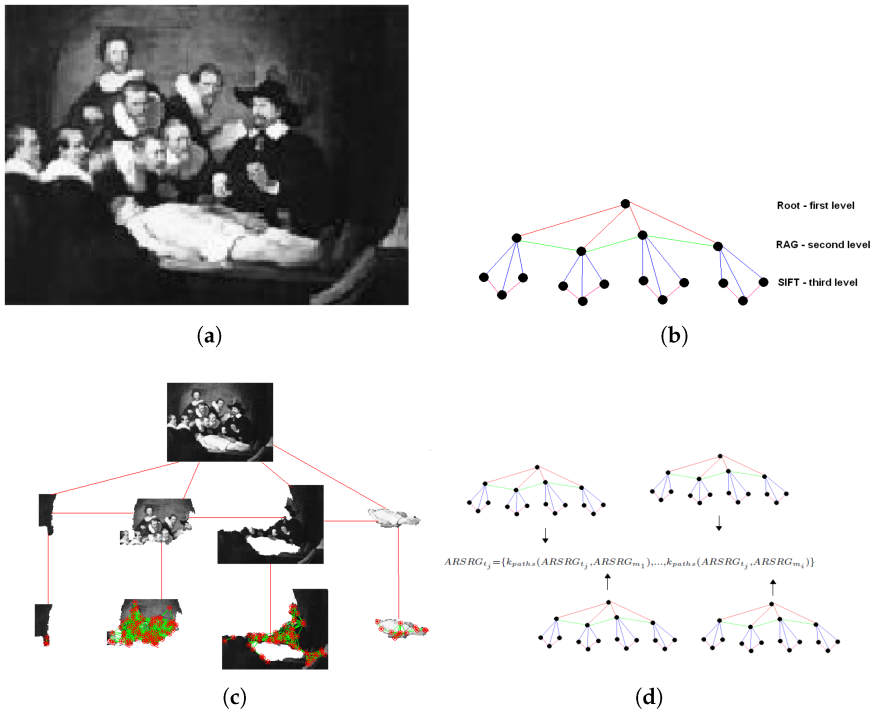
- Graph structure plays an important role in many real-world applications.
- Representation learning on structured data with machine and deep learning methods has shown promising results in different fields.



[Manzo, 2020]

Graph-based data representation

- Reduction and/or transformation of the graph space for better management.
- Many graph embedding methods have been developed aiming at mapping graph data into a vector space.
- Tasks: graph classification, node classification, graph matching, etc.



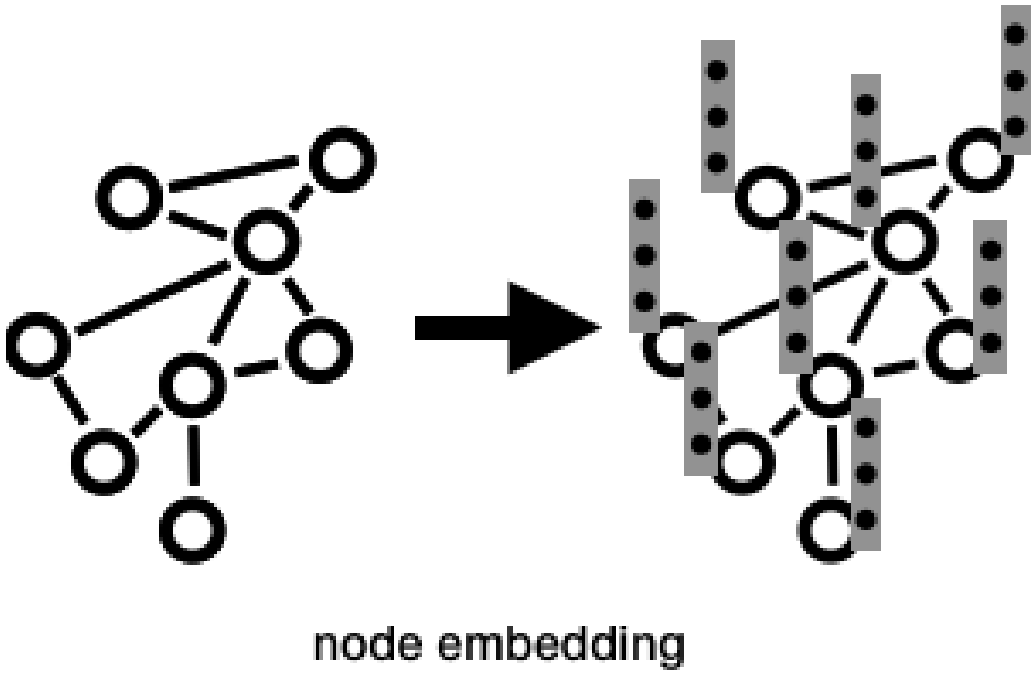
[Manzo, 2019]

Graph embedding

- In a very general definition, graph embedding is any technique that computes or learns a mapping from a graph to a vector space, while preserving relevant graph properties.
- Thus, an embedding produces a d-dimensional feature vector in a novel space, also called *latent space*, trying to preserve the meaningful connection between vertices.

Graph embedding

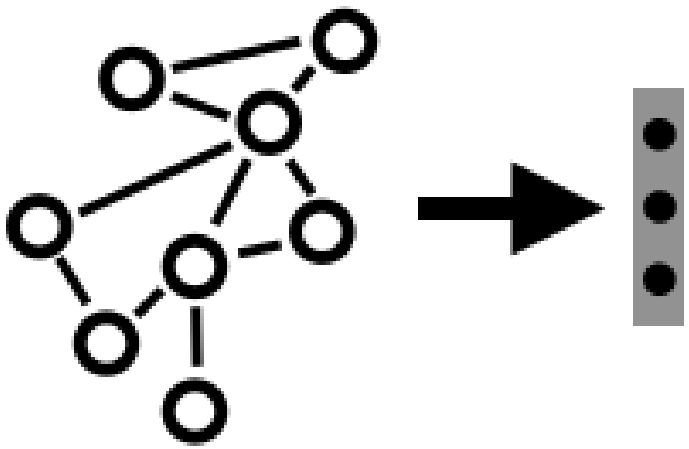
- Given a graph $G{=}(V, E)$, a *graph embedding* (or node-level graph embedding) is a mapping ϕ: $v_i \in V \rightarrow \mathrm{y}_i \in \mathbb{R}^d, i = 1, \ldots, N$, $d \in \mathbb{N}$, such that the function ϕ preserves some proximity measure defined on graph G;



[Manipur et al., 2021]

Graph embedding

- Given a set of graphs $\mathcal{G} = \{G_i\}_{i=1}^{M}$ with the same set of vertices V, a whole-graph embedding is a mapping $\psi \colon G_i \to \mathrm{y}_i \in \mathbb{R}^d, i = 1, \ldots, M$, $d \in \mathbb{N}$, such that the function ψ preserves some proximity measure defined on $\mathcal{G}$;



graph embedding

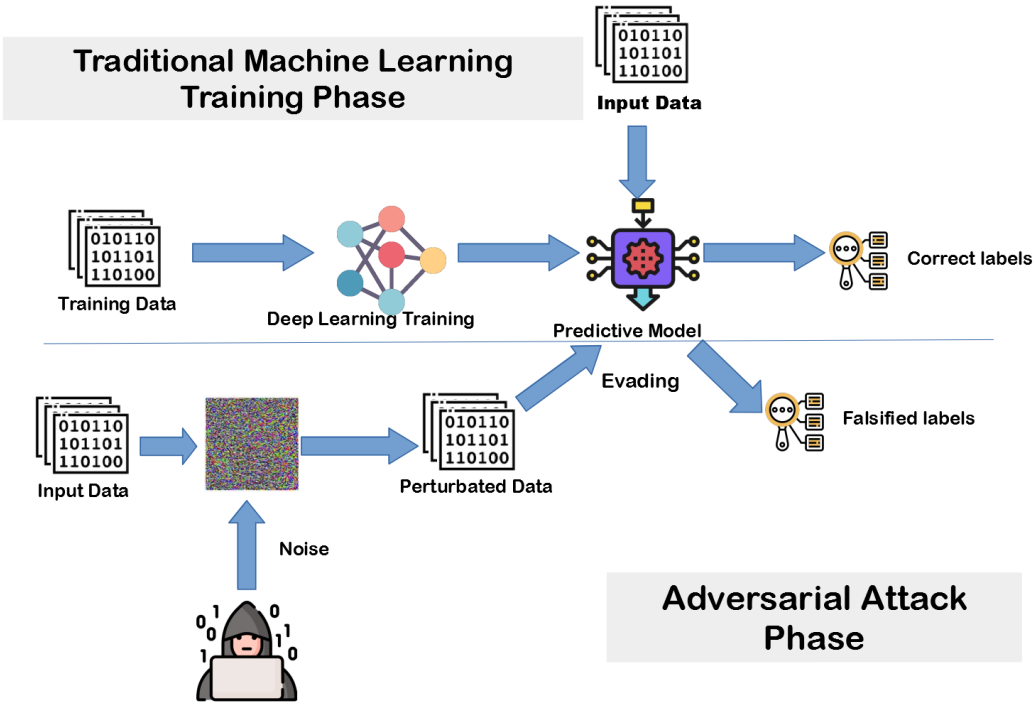
[Manipur et al., 2021]

Graph embedding: critical issues

- Despite the remarkable success, the lack of interpretability and robustness of these models makes them highly risky in fields like biomedicine, finance, and security.
- Typically, sensitive information concerns the user-user relationship within the graph.
- For example, an ill-intentioned person could disguise himself by connecting to other people on a social network.
- Such an "attack" on the model is simple enough but could lead to severe consequences (due to a large number of daily interactions, even if only a few of them are fraudulent, the ill-intentioned could gain enormous benefits).

Adversarial machine learning

- Adversarial machine learning is the area of research in which models vulnerability is studied under adversarial manipulation of their input intended to cause incorrect classification.
- Neural networks and many other machine learning models are highly vulnerable to adversarial perturbations of the input to the model.



[https://morioh.com/p/09872b9d6bde]

Adversarial machine learning

- The focus is on adversarial ML techniques and approaches applied to the graph classification.
- In this domain, the idea is not a scenario in which a "real adversary" intentionally introduces malicious perturbations in the input of learning models.
- The interpretation of "adversarial attacks" within the realm of networks concerns any type of perturbation to the graph structure, due to noise introduced by the experimental environment.

Graph adversarial attacks

- Generally, a network can become damaged through two primary ways: natural failure and targeted attack.
- Natural failures typically occur when a part fails due to natural causes. This results in the malfunction or elimination of a node or edge in the graph.
- Targeted attacks carefully and through precise rules select the nodes and edges of the network for removal in order to maximally disrupt network functionality.

Graph adversarial attacks

- The attention is focused on the modifications to the discrete structures and different attack strategies.
- Generally, the attacker tries to add or delete edges from G to create the new graph.
- These kinds of actions are varied since adding or deleting nodes can be performed by a series of modifications to the edges.

Attack strategies

- Degree-based Attack (DA): a percentage p of graph nodes having the highest degree is removed.
- The degree (or connectivity) δ_{v_i} of node v_i is the number of edges connected to it and can be computed using the graph adjacency matrix $A = \{A_{i,j}\}$ as

$$\delta_{v_i} = \sum_{j \neq i} A_{i,j}.$$

  The effect of a DA is to reduce the total number of edges in the network as fast as possible.
- It only takes into account the neighbors of the target node v when making a decision and can be considered a local attack.
- It is performed with low computational overhead.

Attack strategies

- Betweenness-based Attack (BA): a percentage p of graph nodes having the highest betweenness centrality is removed.
- The betweenness centrality for a node v_i is defined as

$$b_{v_i} = \sum_{j,k \neq i} \frac{\sigma_{j,k}(v_i)}{\sigma_{j,k}},$$

  where $\sigma_{j,k}$ is the total number of shortest paths from node v_j to node v_k and $\sigma_{j,k}(v_i)$ is the number of those paths that pass through the target node v_i.
- It is considered a global attack strategy as the path information is aggregated from the whole network.
- Clearly, global information carries significant computational overhead compared to local attacks.

Experiments

- *Selection of the target nodes and edges for the attack.*
- *Parameters setting.*
- *Robustness.*
- *Vulnerability.*
- *Data driven selection.*

Experiments: target nodes and edges

- Suppose one or a few nodes or edges are perturbed at random.
- In that case, the graph classification results may not change because such a perturbation may not affect or destroy the intrinsic characteristics of graphs discriminating for the classification.
- Finally, nodes and edges target must be chosen according to a criterion.

Experiments: parameters setting

- The choice is undoubtedly difficult as the starting graphs are perturbed. A consequence could also fall on the computational costs during classification.
- As well known, optimizing the parameters is a crucial aspect for obtaining the best performance.
- Concerning this point, the parameter space to choose those that lead to the best results is explored.

Experiments: robustness

- It is always an essential factor in evaluating the performance of the models. In the scenario of adversarial attacks, how to improve the robustness of the classification models?
- It is not certain that, by weakening the structure of the graphs, the transformation into a vector space, through the embedding phase, necessarily produces an unrepresentative features vector, affecting the classification.
- Some methods adapt even when the graph structures are less dense and informative.

Experiments: vulnerability

- It is an essential factor in evaluating the performance of the models. In the scenario of adversarial attacks, how to identify the vulnerability of the classification models?
- Also in this case, by weakening the structure of the graphs, the transformation into a vector space, through the embedding phase, could produce an unrepresentative feature vector, affecting the classification.
- Some methods do not fit when the graph structures are less dense and informative.

Experiments: data driven selection

- The choice of data is driven by the characteristics of the graphs. In this way, models can show robustness or highlight critical issues when a variation of the data occurs.
- Various methods are stressed and chosen for the evaluation based on different characteristics related to data.
- This detail is fundamental for calculating the centrality measures and, therefore, for selecting the nodes to be attacked.

Datasets

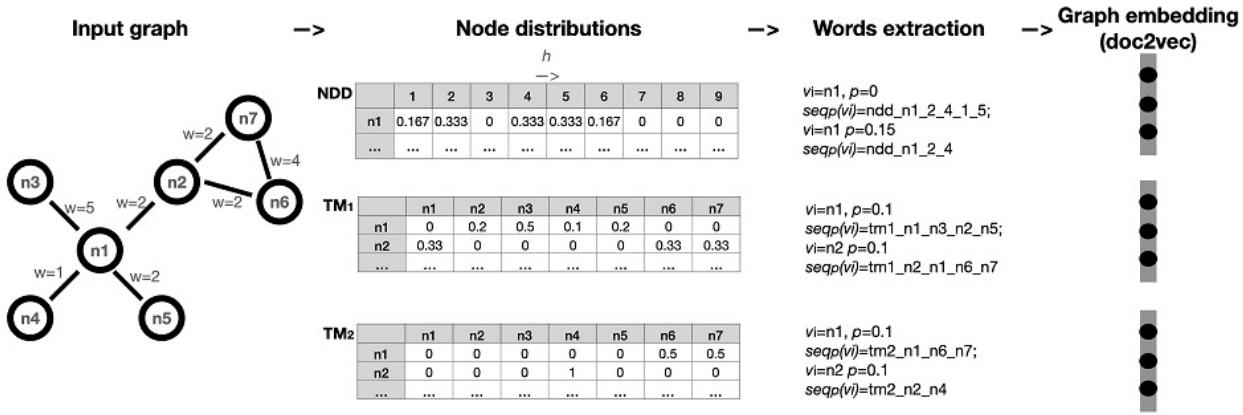
	LFR	MREG	Kidney	Brain fMRI	MUTAG
Graphs	1600	300	299	124	188
Classes	2	3	3	2	2
Samples per class	600/1000	100/100/100	159/90/50	70/54	125/63
Vertices	82	100	1034	263	17.93
Average edges	844.45	1151.71	3226.00	19748.88	39.59
Average edge density	0.13	0.23	0.01	0.57	0.138454
Distinct vertex labels	82	100	1034	263	7
Edge weights	✗	✗	✓	✓	✗
Minimum diameter	3	2	126	0.03	5
Maximum diameter	7	3	455.36	0.07	15
Average degree	20.60	23.03	6.24	150.18	2.19

Compared methods

- **Graph2vec**: provides a Skip-Gram neural network model, typically adopted in the natural language processing domain.
- **GL2vec**: extended version of Graph2vec.
- **IGE**: extracts handcrafted invariant features based on graph spectral decomposition.
- **NetLSD**: computes a compact graph signature derived from the solution of the heat equation involving the normalized Laplacian matrix.
- **FGSD**: provides a graph representation based on a family of graph spectral distances with uniqueness, stability, sparsity, and computational efficiency properties.
- **FeatherGraph**: adopts characteristic functions defined on graph vertices to describe the distribution of node features at multiple scales.

Netpro2vec

- Neural-based embedding framework.
- It looks at basic node descriptions other than the degree, such as those induced by the Transition Matrix and Node Distance Distribution.
- It provides embeddings completely independent from the task and nature of the data.



[Manipur et al., 2021]

Perfomance evaluation

- Results achieved using the original network data (Unattacked), as well as those using data that underwent the removal of the 30% and 50% of the nodes having highest betweenness centrality (BA) or the highest degree (DA) are considered.
- The choice of these percentages p of nodes to be removed aims at investigating the effects of both *moderate* (30%) and *strong* (50%) adversarial attacks.

Relevance feedback

- The performance is evaluated in terms of the Accuracy and the Matthews correlation coefficient (MCC) values:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}$$

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$$

Results: LFR 30% of attacked nodes

Method	Accuracy			MCC		
	Unattacked	BA	DA	Unattacked	BA	DA
GL2Vec	94.59±1.75	84.66±2.45	87.66±2.12	0.88±0.03	0.66±0.05	0.74±0.04
Graph2vec	91.94±2.04	84.41±2.74	89.44±2.13	0.82±0.04	0.66±0.05	0.77±0.04
IGE	**100.00±0.00**	97.06±1.34	97.17±1.17	**1.00±0.00**	0.93±0.02	0.93±0.02
NetLSD	**100.00±0.00**	**99.09±0.73**	99.04±0.71	**1.00±0.00**	**0.98±0.01**	0.97±0.01
FGSD	**100.00±0.00**	97.97±0.99	**99.15±0.68**	**1.00±0.00**	0.95±0.02	**0.98±0.01**
FeatherGraph	**100.00±0.00**	98.99±0.69	99.00±0.74	**1.00±0.00**	0.97±0.01	0.97±0.01
Netpro2vec^{ndd}	**100.00±0.00**	98.41±0.96	97.40±1.16	**1.00±0.00**	0.96±0.01	0.94±0.02
Netpro2vec$^{ndd+tm1}$	**100.00±0.00**	95.26±1.16	72.13±2.99	**1.00±0.00**	0.89±0.03	0.38±0.07

- It is the dataset with the highest number of graphs.
- For the LFR dataset, the performance on unattacked graphs is high for all methods.
- In the case of moderate attacks, NetLSD, FGSD, FeatherGraph and Netpro2vec with NDD respond better to both the types of adversarial attack, showing a lower reduction in Accuracy and MCC values as compared to the other methods.

Results: LFR 50% of attacked nodes

Method	Accuracy			MCC		
	Unattacked	BA	DA	Unattacked	BA	DA
GL2Vec	94.59±1.75	85.36±2.65	83.04±2.56	0.88±0.03	0.68±0.05	0.63±0.05
Graph2vec	91.94±2.04	88.74±2.36	85.51±2.69	0.82±0.04	0.75±0.05	0.70±0.05
IGE	**100.00±0.00**	91.46±2.10	94.17±1.85	**1.00±0.00**	0.81±0.04	0.87±0.03
NetLSD	**100.00±0.00**	93.60±1.93	92.97±1.99	**1.00±0.00**	0.86±0.04	0.85±0.04
FGSD	**100.00±0.00**	77.96±2.54	82.58±2.97	**1.00±0.00**	0.52±0.05	0.62±0.06
FeatherGraph	**100.00±0.00**	**97.17±1.19**	**94.62±1.79**	**1.00±0.00**	**0.93±0.02**	**0.88±0.03**
Netpro2vec^{ndd}	**100.00±0.00**	82.99±2.55	86.67±2.40	**1.00±0.00**	0.63±0.05	0.71±0.05
Netpro2vec$^{ndd+tm1}$	**100.00±0.00**	82.99±2.55	62.99±3.73	**1.00±0.00**	0.63±0.05	0.18±0.08

- It is the dataset with the highest number of graphs.
- For the LFR dataset, the performance on unattacked graphs is high for all methods.
- For stronger attacks, FeatherGraph proves to be the most robust method, experiencing only a slight performance decrease.

Results: Brain fMRI COBRE 30% of attacked nodes

Method	Accuracy			MCC		
	Unattacked	BA	DA	Unattacked	BA	DA
GL2Vec	no conv.	no conv.	no conv.	no conv.	no conv.	no conv.
Graph2vec	43.85±11.27	46.29±13.43	42.58±12.38	-0.18±0.24	-0.09±0.27	-0.18±0.25
IGE	44.88±14.70	48.99±13.33	53.69±14.64	-0.11±0.30	-0.03±0.28	0.05±0.30
NetLSD	56.12±6.59	55.98±12.10	**56.01±8.70**	0.01±0.18	0.09±0.26	0.03±0.21
FGSD	56.54±2.20	54.68±12.75	48.31±13.90	0.00±0.00	0.07±0.26	-0.06±0.29
FeatherGraph	53.77±5.89	52.65±7.77	53.77±12.30	-0.06±0.16	-0.08±0.17	0.01±0.28
Netpro2vec^{ndd}	**56.58±12.74**	58.58±10.20	52.97±11.36	**0.11±0.27**	0.14±0.23	-0.00±0.27
Netpro2vec$^{ndd+tm1}$	**56.58±12.74**	**59.18±13.32**	53.30±12.70	**0.11±0.27**	**0.17±0.28**	**0.05±0.27**

- Netpro2vec, mainly when based on NDD+TM1, appears to be the method that best exploits the network edges' weights.
- It proves to be quite robust to adversarial attacks, experiencing slightly decreased performance for moderate attacks.
- Instead, NetLSD improves its performance when handling moderate DAs, showing the best performance among all the compared methods.

Results: Brain fMRI COBRE 50% of attacked nodes

Method	Accuracy			MCC		
	Unattacked	BA	DA	Unattacked	BA	DA
GL2Vec	no conv.	no conv.	no conv.	no conv.	no conv.	no conv.
Graph2vec	43.85±11.27	51.13±11.93	46.27±13.14	-0.18±0.24	-0.00±0.25	-0.10±0.27
IGE	44.88±14.70	48.96±13.21	56.83±13.16	-0.11±0.30	-0.02±0.27	0.12±0.27
NetLSD	56.12±6.59	50.68±10.30	54.41±6.25	0.01±0.18	-0.08±0.20	-0.02±0.13
FGSD	56.54±2.20	48.49±13.51	45.22±11.42	0.00±0.00	-0.05±0.28	-0.12±0.23
FeatherGraph	53.77±5.89	52.63±12.04	**60.65±13.51**	-0.06±0.16	0.02±0.26	**0.20±0.28**
Netpro2vec^{ndd}	**56.58±12.74**	52.46±13.24	53.83±11.31	**0.11±0.27**	0.02±0.28	0.01±0.26
Netpro2vec$^{ndd+tm1}$	**56.58±12.74**	**56.35±12.46**	53.83±11.31	**0.11±0.27**	**0.11±0.25**	0.01±0.26

- Also in this case , Netpro2vec, mainly when based on NDD+TM1, appears to be the method that best exploits the network edges' weights.
- It proves to be quite robust to adversarial attacks, experiencing slightly decreased performance for strong attacks.
- Instead, Graph2vec improves its performance when handling strong BAs and DAs and the same can be said for FeatherGraph and IGE under strong attack.

Results: Kidney RNASeq 30% of attacked nodes

Method	Accuracy			MCC		
	Unattacked	BA	DA	Unattacked	BA	DA
GL2Vec	90.09±4.74	82.58±6.73	59.83±6.05	0.83±0.08	0.71±0.11	0.25±0.16
Graph2vec	90.79±5.11	79.87±7.05	58.08±5.94	0.83±0.08	0.66±0.12	0.21±0.17
IGE	no conv.	no conv.	no conv.	no conv.	no conv.	no conv.
NetLSD	53.46±7.02	59.07±7.14	62.23±8.68	0.11±0.16	0.25±0.15	0.36±0.15
FGSD	no conv.	no conv.	no conv.	no conv.	no conv.	no conv.
FeatherGraph	81.51±7.96	81.67±6.44	84.36±6.64	0.68±0.13	0.69±0.10	0.74±0.11
Netpro2vec^{ndd}	83.53±6.42	87.22±6.17	85.83±6.19	0.71±0.11	**0.79±0.10**	0.76±0.10
Netpro2vec$^{ndd+tm1}$	**91.27±4.45**	**87.33±5.86**	**90.91±5.60**	**0.86±0.07**	0.79±0.09	**0.85±0.09**

- It is the dataset with the highest number of nodes for each graph.
- IGE and FGSD fail to reach convergence in all the unattacked and attacked cases, yielding no classification model.
- Also in this case, Netpro2vec, mainly when based on NDD+TM1, appears to be the method that best exploits the network edges' weights.

Results: Kidney RNASeq 50% of attacked nodes

Method	Accuracy			MCC		
	Unattacked	BA	DA	Unattacked	BA	DA
GL2Vec	90.09±4.74	73.39±7.56	68.49±7.34	0.83±0.08	0.55±0.12	0.44±0.14
Graph2vec	90.79±5.11	73.02±7.30	67.42±7.44	0.83±0.08	0.54±0.12	0.42±0.14
IGE	no conv.	no conv.	no conv.	no conv.	no conv.	no conv.
NetLSD	53.46±7.02	61.13±8.00	63.27±7.76	0.11±0.16	0.34±0.14	0.38±0.13
FGSD	no conv.	no conv.	no conv.	no conv.	no conv.	no conv.
FeatherGraph	81.51±7.96	81.37±6.83	**89.00±4.79**	0.68±0.13	0.69±0.11	**0.82±0.07**
Netpro2vec^{ndd}	83.53±6.42	87.52±5.66	87.35±5.30	0.71±0.11	0.80±0.10	0.79±0.08
Netpro2vec$^{ndd+tm1}$	**91.27±4.45**	**89.20±5.36**	88.87±5.68	**0.86±0.07**	**0.82±0.09**	0.81±0.09

- It is the dataset with the highest number of nodes for each graph.
- IGE and FGSD fail to reach convergence in all the unattacked and attacked cases, yielding no classification model.
- Also in this case, Netpro2vec, mainly when based on NDD+TM1, appears to be the method that best exploits the network edges' weights.
- Only in the case of DAs, FeatherGraph shows the best performance.

Conclusions and Future Works

- Different whole-graph embedding methods are analyzed and compared to better understand their behavior under adversarial attacks.
- During the attacks, the unique features of each embedding method are analyzed in order to highlight strengths and weaknesses, the latter being varied with respect to the type of attack and dataset.
- In this regard, the robustness of the graph analysis task model is an important issue.
- Future work looks in the directions:
  1. Analysis on different types of datasets and attacks to propose defense mechanisms that can, partially or completely, erase the highlighted limits of existing solutions.
  2. Analysis and exploration of different tasks such as clustering.
  3. Analysis of embedding features adopted for the classification task. Methods like SHapley Additive exPlanations (SHAP) could be applied to learn feature importance and explain the model output.