Data Mining
Programming Project

Project Report
WS 2023/24

Group 4

Adlhart Marlene (01634554)
Friedrich Alexander (11729489)
Mucha Kacper (12043092)

Note: Initially the group consisted of 6 people, however 3 of those dropped out of the course.

Contents

1 Network Data

For analysis, two networks from the KONECT project were chosen - *Yeast* and *Adolescent Health*.

	Nodes	Edges	#Nodes	#Edges	Directed	Weighted
Yeast	proteins	physical interaction	1870	2277	No	No
Adolescent Health	students	friendship	2539	12969	Yes	Yes

Table 1: Basic information on used networks.

1.1 Yeast

This network represents protein-protein interactions in yeast. Nodes represent proteins in yeast and undirected, unweighted edges in the network represent physical interaction between the two proteins. The interaction data represents combined data of multiple experiments, mostly based on two-hybrid screening. The network consists of 1870 nodes and 2277 edges, including self-loops.

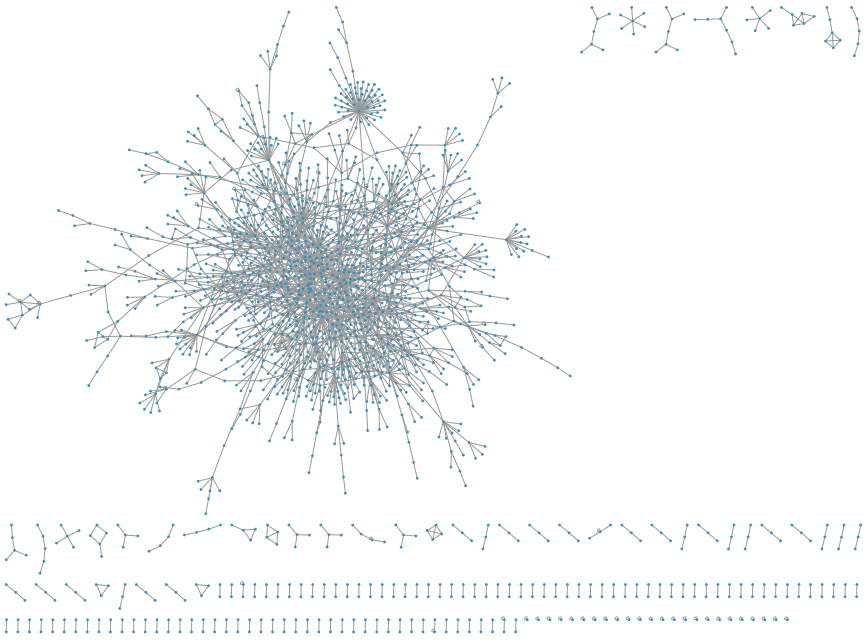

Figure 1: Visualisation of the Yeast network, obtained using *Cytoscape*[1].

[1]Shannon, P. et al., 2003. Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome research, 13(11), pp.2498–2504.

1.2 Adolescent Health

This social network represents friendship among students. The data was obtained from a survey in 1994/1995, where each student should list his 5 best female and his 5 best male friends. Nodes in the network represent students, while directed, weighted edges represent that the student chose the other student as a friend, with edge weights representing the strength of connection. The network consists of 2539 nodes and 12969 edges.

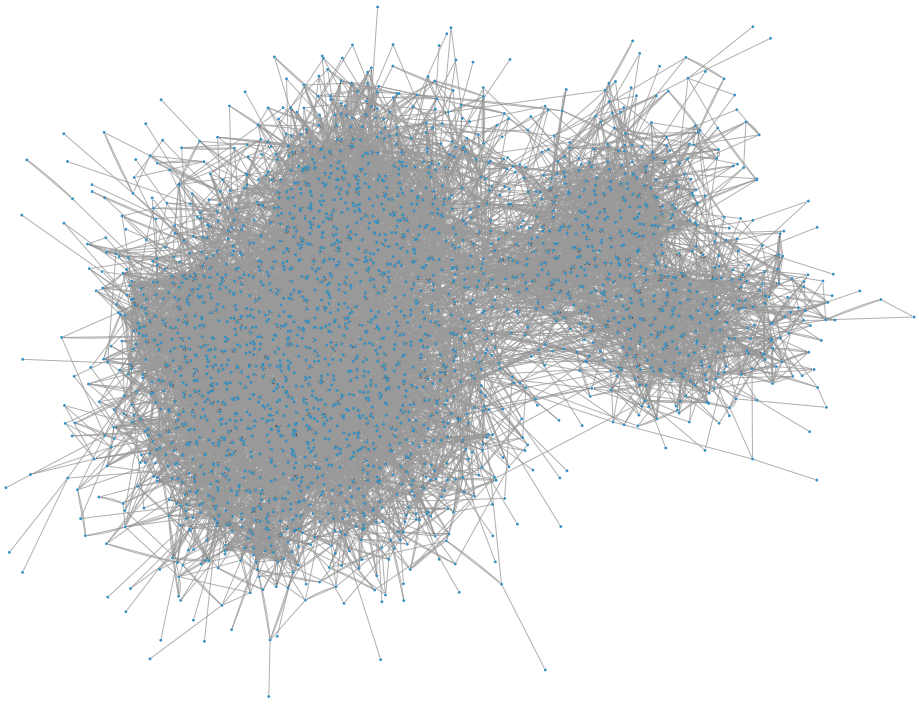

Figure 2: Visualisation of the Adolescent health network, obtained using *Cytoscape*.

2 Network Analytics

2.1 Centrality measures

Degree, betweenness centrality and eigenvector centrality were chosen for analysis of the two networks, using the functionalities of NetworkX. For the weighted Adolescent Health network weights were taken into account for the calculation of all measures. For betweenness centrality, the inverse of the weights were used as a measure of distance, as weights in this data set reflect interaction strength.

When looking at the distribution of the analysed centrality measures for the Yeast network (Figure 3), one can observe that this network is generally characterised by a large number of nodes with low degree, and only few nodes with higher degree. For example, the average degree of nodes in this network is only 2.4, while the protein with the highest degree has 56 interaction partners. The same is true for betweenness and eigenvector centrality, with a distribution heavily skewed towards low values.
In a biological context, these few, highly central proteins could play very fundamental roles in yeast and may be indispensable for the organism. On the other hand, this could mean that the vast majority of proteins could be disturbed, without having drastic effects on the entire organisms. Especially the highly connected proteins may represent interesting target proteins to study. A visualisation of the centrality measures for this network is provided in Figure 5.

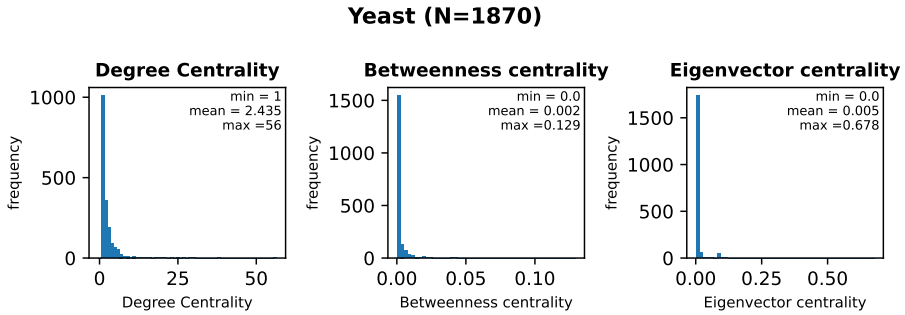

Figure 3: Distribution of degree, betweenness centrality and eigenvector centrality of nodes in the Yeast network.

In contrast to the Yeast network characterised by a small number of highly connected nodes with the majority of nodes having only 1 or 2 connections, the degrees of nodes in the Adolescent Health network are more evenly distributed with a large number of nodes having an intermediate degree, close to the average (Figure 4). Instead of having only a few highly connected individuals, the majority of the people in the network seem to have more similar connections. On the other hand, the betweenness centrality is also heavily skewed towards 0. Nodes with a high betweenness centrality could e.g provide bridges between

different friend groups. Only a few nodes which show a high eigenvector centrality could be identified. These people could have a large influence on the entire network, as they are connected to highly influencial people. A visualisation of the centrality measures for this network is provided in Figure 6.

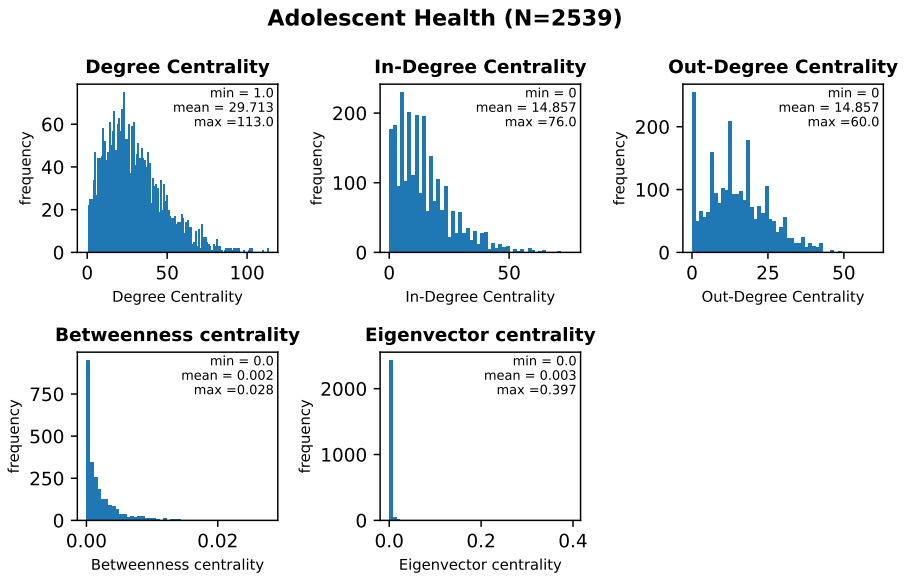

Figure 4: Distribution of degree, betweenness centrality and eigenvector centrality of nodes in the Adolescent Health network. Distribution for in and out degree is shown separately, with distribution of degree centrality as sum of in and out degree for a node shown as well.

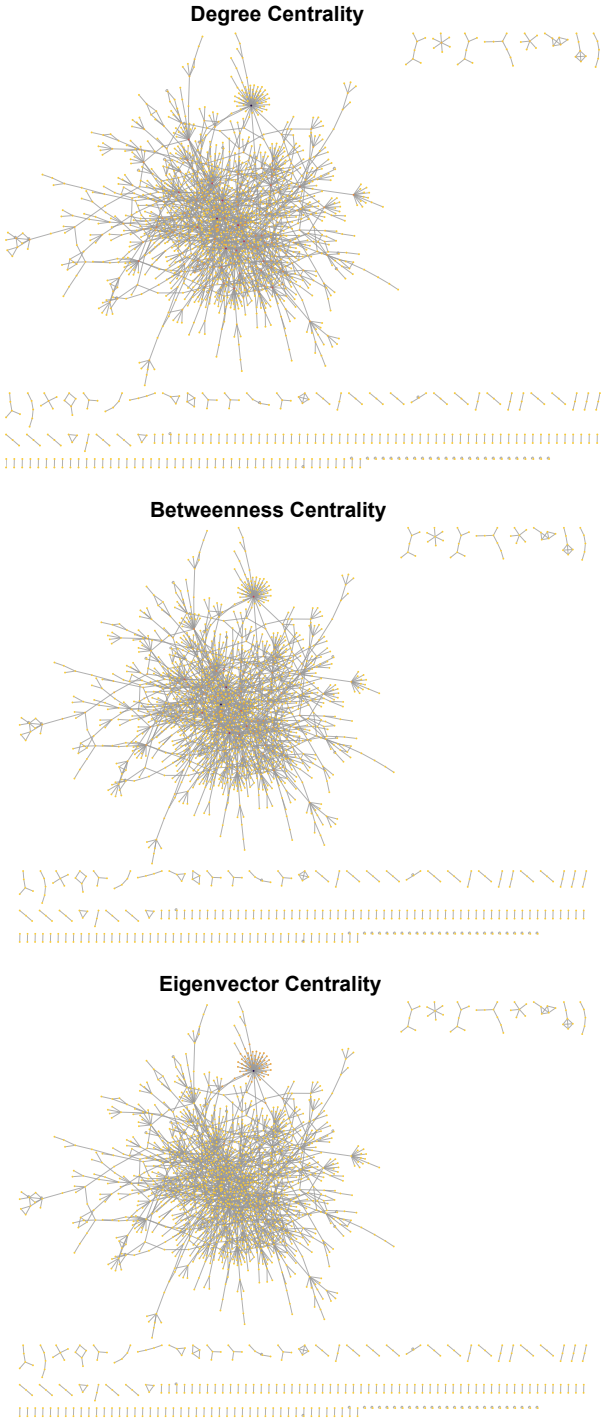

Figure 5: Visualisation of the Yeast network, with nodes colored by degree, betweenness centrality and eigenvector centrality. Obtained using *Cytoscape*.

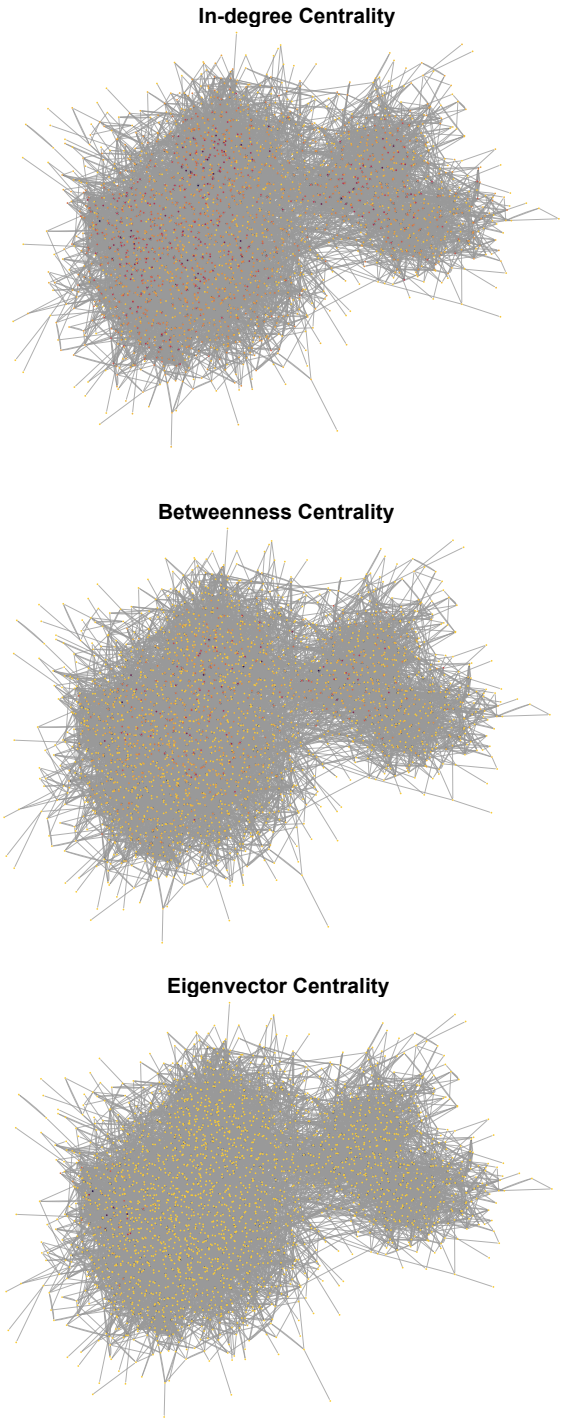

Figure 6: Visualisation of the Adolescent Health network, with nodes colored by In-degree, betweenness centrality and Eigenvector centrality. Obtained using *Cytoscape*.

2.2 Comparison of ranking of nodes for weighted and unweighted networks

The obtained ranking of the nodes for the two networks was compared to the ranking of the nodes obtained for the unweighted/weighted version of the graph, using Kendall's τ coefficient (Table 2). For the unweighted yeast network, random integer weights between 1 and 20 were assigned to all edges for this task.

	Yeast	Adolescent Health
Degree centrality	0.69	0.74
In-Degree centrality	-	0.8
Out-Degree centrality	-	0.69
Betweenness centrality	0.88	0.69
Eigenvector centrality	0.83	0.52

Table 2: Kendall's τ coefficient between ranking of nodes obtained using weights and not using weights of edges for the calculation of centrality measures.

While generally an intermediate to strong correlation (Kendall's τ of approximatley 0.5-0.9) was observed for the rankings of the different centrality measures and networks, the experiment shows that taking edge weights into account can have a significant impact on the obtained node rankings, depending on the centrality measure and the network structure itself.

2.3 Running time

	Yeast		Adolescent Health	
	unweighted	weighted	unweighted	weighted
Degree centrality	0.000003	0.000007	0.000012	0.000024
In-Degree centrality	-	-	0.000002	0.000026
Out-Degree centrality	-	-	0.000002	0.000020
Betweenness centrality	6.041188	11.929196	19.443871	43.256149
Eigenvector centrality	0.390676	1.122657	0.366255	0.895594

Table 3: Running time in seconds, for each centrality measure and network and with or without including the (random) edge weights in the calculation.

3 Learning Node Properties

In this section we discuss the capability of graph neural networks to learn classical node properties and centrality measures in a supervised learning approach. The target values we want to predict are the degree of the node, its Eigenvector centrality and Page Rank.

3.1 Approach and Method

To explore this question, we use the networks discussed in the previous exercises, calculate the corresponding target values, degree centrality for the (in-)degree, Eigenvector centrality and Page Rank, and then trained separate graph neural networks for each task. All of the target values have been standardized to the interval [0,1] to simplify comparability. Additionally we randomly generated three dimensional feature vectors for each node according to a uniform distribution in [0,1], as we found that a higher dimensional feature vector yielded better results.

The model used, consists of GINConv layers, each including a neural network block, made up of a linear layer, a batch normalization layer, a non-linear activation function (ReLU), another linear layer and another non-linearity (ReLU). Finally there are two more linear layers with a non-linearity (ReLU) in between, that produce the structured output.

Training utilizes the Adam optimizer and is based on L1-loss. We utilize a learning rate of 0.001 together with weight decay in conjunction with early stopping to prevent excessive overfitting noticed in early trials. We test different configurations, varying the hidden layer dimension from 32 to 256 and the number of graph isomorphism network blocks from 3 to 5, which are trained for up to 1000 epochs each. The best model is then chosen based on the evaluation on the test set. We apply a train mask, that reduces the amount of the data for calculation of gradients during training to 90% of the original data and the test error is then calculated based on the other 10% of the graph data, the model has not seen previously. The same masking is also applied to calculate the values of Kendall's Tau.

3.2 Results and Analysis

3.2.1 Train and test error

Network	Centrality measure	Best Config	Final Train error	Minimum Test error
Adolescent health	In-Degree Centrality	128_4	0.00483	0.01113
	Eigenvector Centrality	32_5	0.00153	0.00306
	PageRank	128_5	0.02326	0.02842
Yeast	Degree Centrality	128_3	0.00236	0.00416
	Eigenvector Centrality	256_3	0.00068	0.00100
	PageRank	256_5	0.00205	0.00348

Table 4: Final train and test error (l1 loss, i.e mean absolute error) for the best obtained configuration for the two networks and different centrality measures. The first value in the column *Best Config* corresponds to the hidden layer dimension, while the second value corresponds to the number of graph isomorphism blocks, of the chosen configuration.

In general, the obtained train and test errors have similar orders of magnitude within all networks and centrality measures (Table 4). All of them being relatively low, indicating that the model has successfully learned the patterns of the data. However, we noticed that for all models the test error is higher than the train error, indicating slight overfitting to the training data.

3.2.2 Absolute error of predictions

The distribution of the absolute errors between the value predicted by the neural network and the true value for all three centrality measures is depicted in Figure 7 for the Yeast network and Figure 8 for the Adolescent Health network, together with scatter plots of the predicted vs. true values.

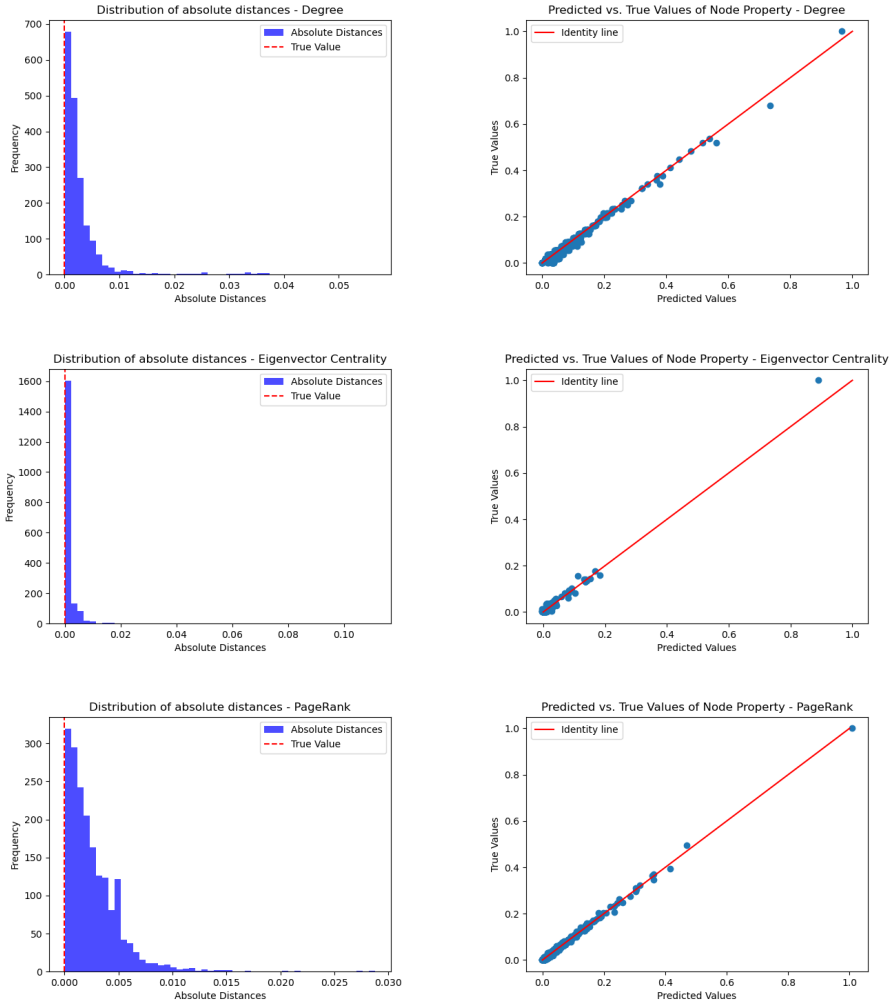

Figure 7: Model prediction performances of Yeast network. In the left column, the distribution of absolute errors between true and predicted values is given, while in the right column a scatter plot of true vs. predicted values is shown for degree, eigenvector centrality and page rank.

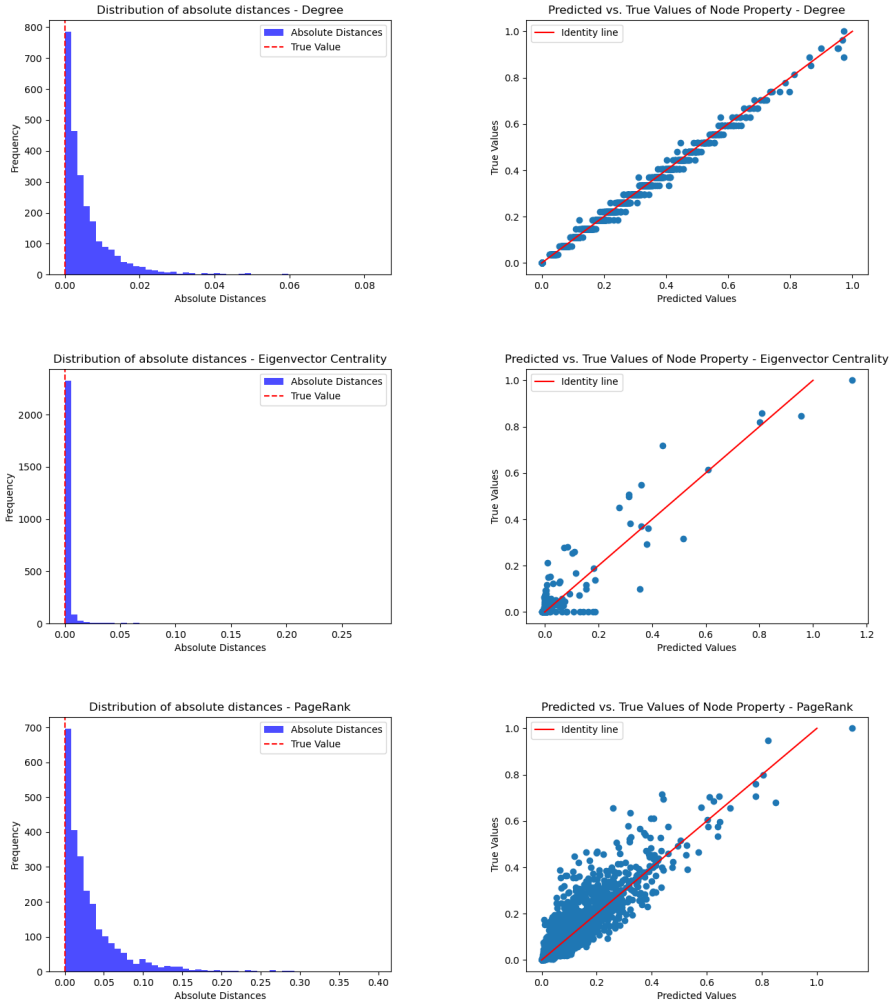

Figure 8: Model prediction performances of Adolescent health network. In the left column, the distribution of absolute errors between true and predicted values is given, while in the right column a scatter plot of true vs. predicted values is shown for in-degree, eigenvector centrality and page rank.

In general, the absolute errors are skewed towards 0 in all cases, indicating a mostly good agreement between the prediction and true values, also apparent when examining the scatter plots. In case of Degree centrality, for both networks the prediction generally closely matches the true values, with most errors below 0.015 for Yeast and 0.04 for Adolescent Health.
A similar situation is observed for the Yeast network regarding eigenvector centrality and page rank, where most predictions are generally good, with the

exception of the prediction of eigenvector centrality of the single, most central node. On the other hand, for the Adolescent Health network, there is a much wider spread as can be seen in the scatter plots, but still a clear correlation between predictions and true values.

3.2.3 Kendall's Tau

The ranking of the nodes in the test set was compared to the obtained ranking of the nodes from the prediction of the GNNs, using Kendalls τ coefficient (Table 5).
As anticipated from Figure 7 and 8, a strong correlation between the rankings is obtained for in-degree centrality in case of the Adolescent Health network (Kendalls τ=0.94) and although not as good, still a moderate to strong correlation is obtained in case of degree centrality for the yeast network (Kendalls τ=0.76). Similarly a moderate to strong correlation was observed for Page Rank in both networks. On the other hand, the node rankings obtained from eigenvector centrality do not strongly agree with the true rankings, with kendalls τ of 0.33 and 0.56, for the Adolescent Health and Yeast network respectively. Observing these measurements in conjunction with the previous figures, we conclude that the regression models struggled to reproduce correct orderings when confronted with many nodes that share a very similar centrality value, as there are no induced similarities in their randomly generated features that the model could identify.

Network	**Centrality measure**	**Kendall's tau**
Adolescent health	In-Degree Centrality	0.94
	Eigenvector Centrality	0.33
	PageRank	0.76
Yeast	Degree Centrality	0.76
	Eigenvector Centrality	0.56
	PageRank	0.88

Table 5: Kendalls τ ceofficient between the ranking of the nodes in the test set obtained from their true and predicted values.

3.2.4 Running Time of GNN vs. classic centrality algorithms

In general, the running time of the graph neural network based method was faster (in the range of approximately 1 to 600 ms) as compared to the centrality measure algorithms of NetworKit, with the only exception of page rank in the Yeast network (Table 6). These results were obtained running our models on a T4 GPU via Google Colab, which significantly improved prediction times for the GNNs.

Network	Centrality measure	Time [ms] NetworKit	Time [ms] GNN	Time Difference [ms]
Adolescent health	In-Degree Centrality	11.87	1.95	9.92
	Eigenvector Centrality	104.43	2.45	101.98
	PageRank	3.68	2.48	1.2
Yeast	Degree Centrality	14.73	1.46	13.28
	Eigenvector Centrality	608.34	1.61	606.72
	PageRank	2.34	2.53	-0.19

Table 6: Running times of the classic centrality algorithms of NetworKit and the graph neural network based method (ignoring the time for training) as well as their difference, for the two networks and all centrality measures, in milliseconds.

3.3 Conclusion

In our implementation of the GIN architecture we found that, given the models we used, the performance of neural networks to approximate degree, Eigenvector centrality and Page Rank is almost always faster than the classical NetworKit implementations, while producing similar results.

As absolute errors are small and the trends show positive correlation, as can be seen in figure 7 and figure 8, as well as mostly high values for Kendall's tau, see Table 5, we conclude that, even though we were unable to achieve perfect accuracy, the neural network approach has its advantages. Using it as a preliminary filtering, to reduce the number of nodes to apply the classical algorithms to, or to get a general overview seems promising.