Mara Mattes – mattes@dice.hhu.de

Collusion Detection with Graph Neural Networks

Lucas Gomes[a], Jannis Kück[b], Mara Mattes[b], Martin Spindler[a], Alexey Zaytsev[c]

[a]Statistics Institute, BWL, Universität Hamburg, Hamburg, Germany;
[b]Düsseldorf Institute for Competition Economics, Heinrich Heine University Düsseldorf, Düsseldorf, Germany;
[c]Skolkovo Institute of Science and Technology (Skoltech), Moscow, Russia.

Summary

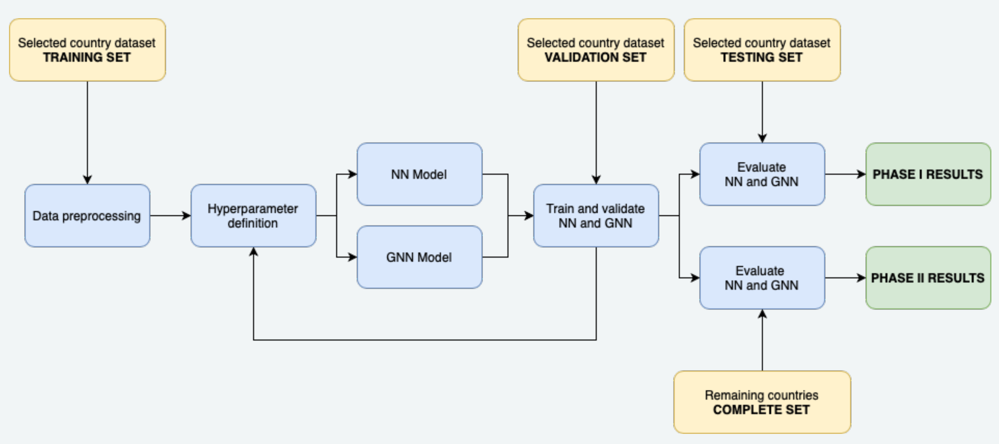
- Predicting collusion patterns across different international markets with Feed-Forward Neural Networks (NNs) and Graph Neural Networks (GNNs).
- Phase I: Training individual models tailored to specific datasets representing diverse markets, including Japan, the United States, Switzerland, Italy, and Brazil.
- Phase II: Focuses on evaluating the predictive power and generalizability of the trained models by applying them to new, unseen datasets from other markets/regions, incorporating Out-of-Distribution generalization to assess their performance in detecting collusion patterns in datasets they were not originally trained on.
- We highlight the GNNs' enhanced ability to detect complex collusive patterns more effectively than NNs.

Data

Empirical analysis is based on bidding data of public tenders from Japan, Brazil, Italy, St. Gallen & Graubünden, Ticino (both Switzerland) and the US. The datasets are from Rodríguez et al. (2022).

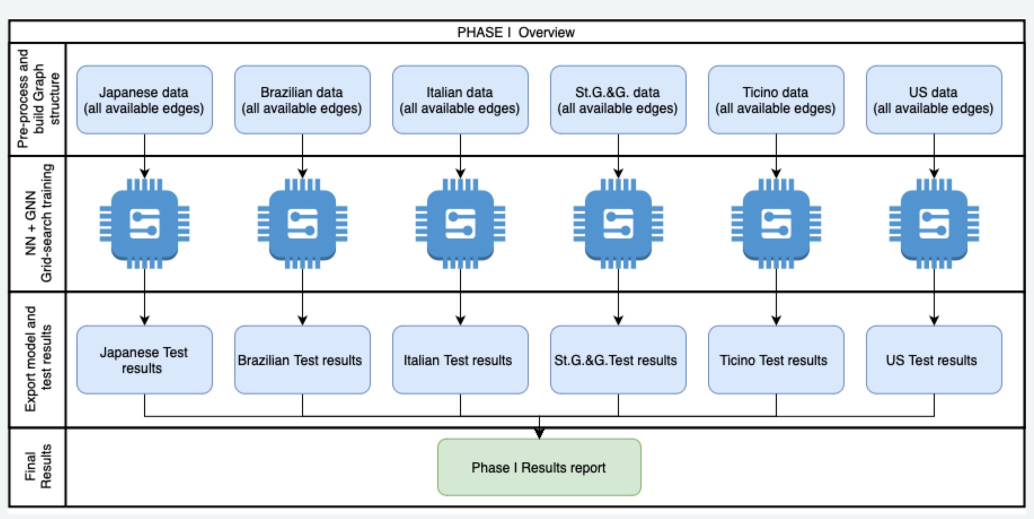
Method Overview

We consider a binary classification problem where we want to classify each bid in a tender as either collusive or non-collusive.



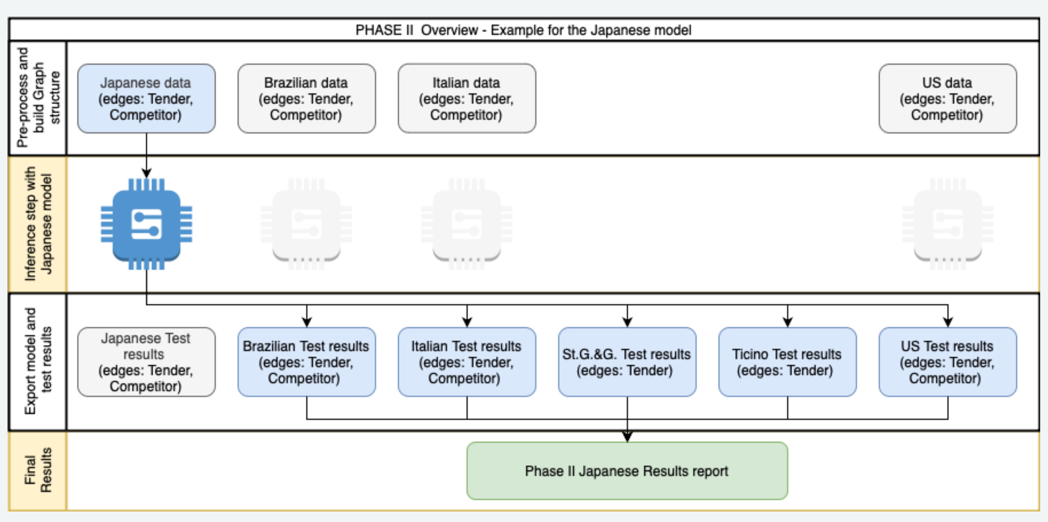
- Phase I: Collusion Prediction in Single Markets

Train one model for each country/region (Japan, Brazil, Italy, St. Gallen and Graubünden, Ticino, US) and then evaluate performance on the respective testing set of the same country/region.



- Phase II: Transportability

Train one model for each country (Japan, Brazil, Italy, US) and then evaluate the performance on the remaining testing sets of the other countries/regions.



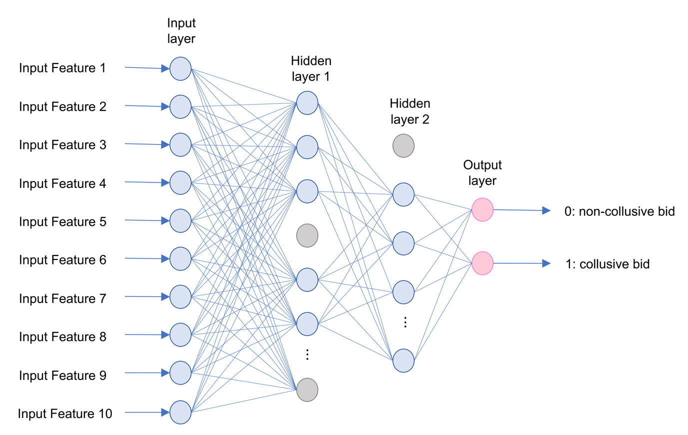
Detection Method

We predict collusive bids in different international markets using NNs and GNNs. GNNs are tailored to exploit the network structure present in bid rigging. Therefore, we aim to show that GNNs can detect collusion patterns more effectively than NNs.

Feature	Description
Bid_value	Price offered by the company
$Number\_bids$	Total number of bids of a given Tender
$Winner$	Binary variable: 1 if bid is winner of the Tender, 0 otherwise
CV	Coefficient of Variation
SPD	Spread inside Tender
$DIFFP$	Difference between the two lowest bids within a Tender
RD	Relative Distance
$KURT$	Excess Kurtosis
$SKEW$	Skewness
$KSTEST$	Kolmogorov-Smirnov test

Table 1: Features for NN and GNN

- Feed-Forward Neural Network



- Relational Graph Convolutional Network

Relational Graph Convolutional Networks (Schlichtkrull et al., 2018) are an extension of Graph Convolutional Networks that handle graphs with multiple types of relations between nodes.

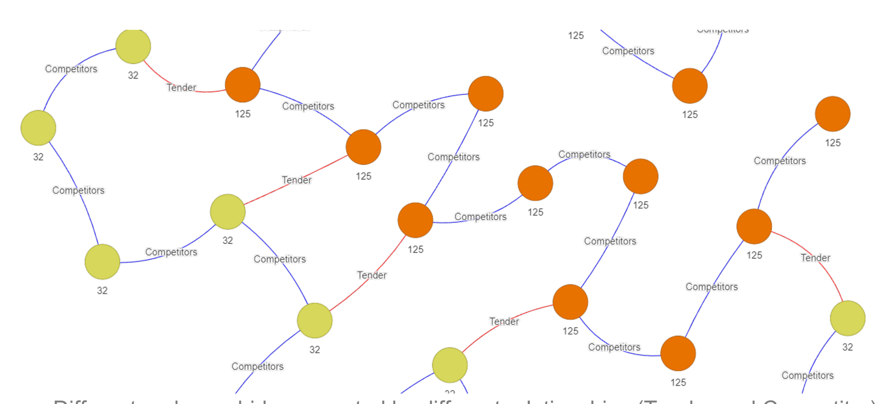


Figure: Different nodes as bids connected by different relationships (Tender and Competitor)

A graph G with $\mu \in \{1, \ldots, m\}$ nodes and $\hat{f} \in \{1, \ldots, f\}$ feature dimensions has its raw input feature vectors β_μ represented as a matrix X of features $x_{\mu,\hat{f}}$ with dimensions $m \times f$:

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,f} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,f} \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}$$

The number of layers is represented as $k \in \{0, \ldots, K-1\}$.
The representation of the first layer is

$$h_\mu^{(k=0)} = [x_{\mu,1} \ \ldots \ x_{\mu,f}] = \beta_\mu \quad \forall \mu.$$

The representation of the layers $k \in \{1, \ldots, K-1\}$ is

$$h_\mu^{(k+1)} = \sigma\left(\sum_{r \in R} \sum_{v \in \mathrm{N}_\mu^r} \frac{1}{d_{\mu,r}} W_r^{(k+1)} h_v^{(k)} + W_0^{(k+1)} h_\mu^{(k)} \right) \forall k, \forall \mu,$$

where N_μ^r denotes the set of neighbor indices of node μ under edge relation $r \in R$. $\sigma(\cdot)$ is an activation function. $d_{\mu,r}$ is the degree of node μ under relation r and a normalization constant. $W_r^{(k+1)}$ is a weight matrix for each relation type r and $W_0^{(k+1)}$ is a weight matrix for self-connections.

In the context of this work, the nodes are bids. The features of the GNN model are the same as those of the NN model and we have four different edge relation types.

Edge relations	Description
Tender	Tender ID, connect bids made in the same Tender
Competitor	Company ID, connect bid with previous bid and next bid made by the same Company
Location	Tender Geographical Location ID, connect bids with the same Tender Location ID
Site	Company Geographical Site ID, connect bids with the same Company Site ID

Table 2: GNN edges between nodes

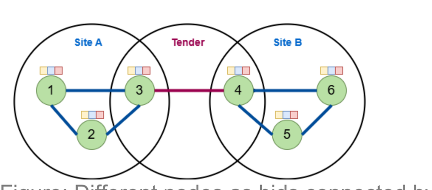


Figure: Different nodes as bids connected by different relationships (Tender and Site)

Dataset	Tender ID	Company ID	Location ID	Site ID
Japan	☒	☒	☒	☐
Brazil	☒	☒	☒	☐
America (US)	☒	☒	☒	☐
Italy	☒	☒	☐	☒
St. Gallen and Graubünden	☒	☐	☐	☐
Ticino	☒	☐	☐	☐

Table 3: Edges available to each dataset

Results

- Phase I

Trained on	Japan		Brazil		Italy	
Mean (SD)	NN	GNN	NN	GNN	NN	GNN
F1-score	0.41 (0.04)	**0.64 (0.05)**	0.67 (0.04)	**0.82 (0.05)**	0.54 (0.05)	**0.56 (0.08)**
Balanced accuracy	0.81 (0.02)	**0.89 (0.02)**	0.85 (0.02)	**0.93 (0.02)**	0.60 (0.02)	**0.63 (0.06)**
Precision	0.27 (0.04)	**0.51 (0.08)**	0.54 (0.05)	**0.74 (0.08)**	0.45 (0.05)	**0.55 (0.07)**
Recall	0.79 (0.06)	**0.86 (0.06)**	0.90 (0.03)	**0.94 (0.05)**	**0.68 (0.08)**	0.59 (0.11)

Trained on	St.G&G		Ticino		US	
Mean (SD)	NN	GNN	NN	GNN	NN	GNN
F1-score	**0.67 (0.07)**	0.63 (0.03)	0.93 (0.03)	**0.98 (0.01)**	0.23 (0.12)	**0.37 (0.13)**
Balanced accuracy	0.54 (0.03)	**0.61 (0.02)**	0.85 (0.04)	**0.98 (0.00)**	0.55 (0.05)	**0.64 (0.10)**
Precision	0.63 (0.04)	**0.69 (0.02)**	0.95 (0.01)	**1.00 (0.00)**	**0.41 (0.31)**	0.35 (0.15)
Recall	**0.74 (0.18)**	0.59 (0.05)	0.91 (0.04)	**0.97 (0.01)**	0.35 (0.20)	**0.47 (0.25)**

Table 4: Test results for training within countries

- F1-score: GNN models have outperformed the NN in all analyzed scenarios, except for St. G&G (model with the fewest available edges).
- Differences in the performance of NN and GNN are smaller in models where only one edge is available (St.G&G, Ticino).
- However, in some countries (especially in the US, but also in Italy and St.G&G), the performance of both NN and GNN is low.

- Phase II

Trained on Japanese data

Inference on	Brazil		Italy		US		St.G&G		Ticino	
Mean (SD)	NN	GNN	NN	GNN	NN	GNN	NN	GNN	NN	GNN
F1-score	**0.06**	0.00	0.08	**0.23**	0.00	**0.08**	0.00	**0.26**	0.07	**0.62**
Balanced accuracy	**0.52**	0.50	**0.51**	0.46	0.50	**0.52**	0.5	**0.55**	0.52	**0.73**
Precision	1.00	1.00	**0.46**	0.32	**1.00**	0.32	**1.00**	0.78	1.00	1.00
Recall	**0.03**	0.00	0.04	**0.18**	0.00	**0.05**	0.00	**0.16**	0.04	**0.45**

Table 5: Inference with model trained on Japanese data

Trained on Brazilian data

Inference on	Japan		Italy		US		St.G&G		Ticino	
Mean (SD)	NN	GNN	NN	GNN	NN	GNN	NN	GNN	NN	GNN
F1-score	**0.18**	0.02	**0.08**	0.03	0.09	**0.26**	**0.73**	0.31	0.00	**0.03**
Balanced accuracy	**0.60**	0.41	**0.51**	0.50	0.45	**0.52**	**0.54**	0.47	**0.49**	0.47
Precision	**0.10**	0.02	**0.46**	0.41	0.07	**0.15**	**0.61**	0.53	**1.00**	0.49
Recall	**0.88**	0.04	**0.05**	0.01	0.10	**0.92**	**0.89**	0.22	0.00	**0.01**

Table 6: Inference with model trained on Brazilian data

Trained on Italian data

Inference on	Japan		Brazil		US		St.G&G		Ticino	
Mean (SD)	NN	GNN	NN	GNN	NN	GNN	NN	GNN	NN	GNN
F1-score	0.13	**0.16**	**0.37**	0.31	0.22	**0.25**	**0.70**	0.08	0.43	0.43
Balanced accuracy	0.46	**0.54**	**0.60**	0.53	0.50	0.50	0.48	**0.49**	0.46	0.46
Precision	0.07	**0.09**	**0.22**	0.20	0.14	0.14	**0.58**	0.50	0.78	0.78
Recall	**0.78**	0.54	**1.00**	0.63	0.48	**1.00**	**0.87**	0.05	0.29	**0.30**

Table 7: Inference with model trained on Italian data

Trained on US data

Inference on	Japan		Brazil		Italy		St.G&G		Ticino	
Mean (SD)	NN	GNN	NN	GNN	NN	GNN	NN	GNN	NN	GNN
F1-score	**0.16**	0.15	0.12	**0.15**	0.55	**0.57**	0.60	**0.64**	0.84	**0.87**
Balanced accuracy	**0.53**	0.51	0.25	**0.31**	**0.51**	0.50	0.55	**0.57**	**0.50**	0.47
Precision	**0.09**	0.08	0.07	**0.10**	0.40	0.40	0.64	0.64	**0.82**	0.64
Recall	0.79	**1.00**	0.27	**0.33**	0.89	**0.97**	0.57	**0.64**	**0.86**	0.64

Table 8: Inference with model trained on US data

- Trained on Japanese data: Ticino test data shows the best performance for the GNN model, while NN model does not perform well across all countries.
- Trained on Brazilian/Italian data: St.G&G test data on NN model almost same performance as training on St.G&G data, while performance on GNN model is lower. All countries show rather low performance with balanced accuracy of approx. 50%.
- Trained on US data: St.G&G test data on NN and GNN model almost same performance as trained on St.G&G data.
- Overall, rather low performance of NN and GNN models across almost all countries. It seems that the individual markets are too different in terms of their collusive patterns.

Conclusion

- We show that GNNs are beneficial to predict collusion and can outperform standard Neural Networks.
- This work provides insights when GNNs are in particularly useful.
- The type of the tender seems to be crucial for modeling collusion detection (e.g. tenders for procurement of school milk are less complex and fewer companies participate in these tenders).
- Training of NN and GNN models in one country and testing in another country with rather poor results.

References

Rodríguez, M. J. G., Rodríguez-Montequin, V., Ballesteros-Pérez, P., Love, P. E., & Signor, R. (2022). Collusion detection in public procurement auctions with machine learning algorithms. Automation in Construction.

Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M., 2018. Mod- eling relational data with graph convolutional networks, in: The Semantic Web: 15th Inter- national Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15, Springer. pp. 593–607.