

# Link Prediction Method in Graph Objects by Auto Encoding in Graph Neural Networks

Vladyslav Shlianin<sup>1</sup>, Yuri Gordienko<sup>1</sup> and Sergii Stirenko<sup>1</sup>

<sup>1</sup>National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37 Peremohy Aveniu, 03056, Kyiv, Ukraine

## Abstract

Link prediction problem is significant for a better understanding of the hidden or lost connections between objects in hierarchical structures like networks, for example, in social, business, biological, medical, and other domains. Recently, Graph Autoencoders (GAE) and Variational Graph Autoencoders (VGAE) deep neural networks (DNNs) emerged as effective tools for resolving various problems. In this paper, their variations were used to solve the link prediction problem for graph objects in the legal document context. For this purpose, the customized dataset in the shape of the hierarchical set of Ukrainian legal acts adopted by the Ukrainian parliament (Verkhovna Rada of Ukraine) and Ukrainian government (the Cabinet of Ministers of Ukraine) was constructed, and its exploratory data analysis (EDA) was performed. Several GAE and VGAE models were proposed and applied for the dataset, the comparison analysis was performed for all of the models considered, and a conclusion was made as to possible further improvements of the method proposed for other real-world graph data in various domains.

## Keywords

Neural networks, deep learning, graph, graph neural networks, autoencoder, graph autoencoding, link prediction

## 1. Introduction

A link prediction task is a prediction of the availability of a link between two nodes in a network. Many examples of link prediction can be found in various everyday applications like a search of friendship connections between users in social networks, estimation of potential business connections between companies in markets, prediction of gene-protein or protein-protein interactions in biological networks, etc [1, 2].


Traditionally, the link prediction problem is solved by assuming that the more similar nodes in a graph, the more likely they are to have edges [3]. In these approaches, link prediction is calculated by investigating the similarity between nodes in a graph, taking into account information about a graph topology. However, not all relations in real-world graphs are based on similarity. For instance, in some graphs like the legal acts network (see details below), connections are based on auxiliary references and facts about the availability of other legal acts.


---

*MoMLeT+DS 2022: 4th International Workshop on Modern Machine Learning Technologies and Data Science, November, 25-26, 2022, Leiden-Lviv, The Netherlands-Ukraine.*

✉ vladyslav.shlianin@gmail.com (V. Shlianin); yuri.gordienko@gmail.com (Y. Gordienko); sergii.stirenko@gmail.com (S. Stirenko)

ORCID 0000-0003-3833-4957 (V. Shlianin); 0000-0003-2682-4668 (Y. Gordienko); 0000-0002-9395-8685 (S. Stirenko)

 © 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

With the advancement of deep neural networks (DNNs) and graph neural networks (GNNs), graph autoencoders (GAEs) and variational graph autoencoders (VGAEs) [4] have been proposed to learn graph embeddings in an unsupervised way. It has been shown that these methods are effective for link prediction tasks. It is worth noting that GAEs and VGAEs mostly rely on graph convolutional networks (GCN) to encode nodes [5].

Such approach works well both on homogeneous and heterogeneous graphs, also known as knowledge graphs [6]. One of the notable examples even performs a study on the knowledge graph of Austrian judicial and legal acts [7]. However, this example implements preexisting models, such as Word2Vec and Doc2Vec.

Usually, to get data about links between legal acts, Ukrainian lawyers have to visit the official parliament portal to get it there. However, the data entry process is not automated and, therefore, incomplete, especially for codified laws and older legal acts. The link prediction method described in this paper aims to reduce this problem by helping lawyers and data entry specialists to get more accurate representations of links between acts with the highest possible precision. In addition to this practical use case, the link prediction problem generally has many real-life applications, from modelling recommendation systems to predicting user interactions in social networks. Recently, link prediction problem appeared in numerous medical applications like disease-gene association prediction problem [8] and other critically important medical problems related to cancer disease diagnostics, and treatment [9, 10].

In this paper, we focus on the link prediction problem, particularly in the context of a hierarchy of Ukrainian legal acts relations. To solve this problem, some homogeneous graphs can be constructed and experimentally compared by application of different GAE and VGAE models with subsequent performance comparison.

The paper has the following structure: section 2. *Background and Related Work* contains a short outline of similar attempts to use various GCNs to investigate the link prediction problem, section 3. *Methodology* presents the dataset, structure of DNNs, and metrics used, section 4. *Experimental* describes the results obtained, section 5. *Discussion* gives the analysis of the methods used, and section 6. *Conclusions* proposes a summary of the further improvements.

## 2. Background and Related Works

Recently, GCN have been studied to extend the possibilities of neural networks on working with data, represented as a graph. Designing a convolutional operator is a key issue and can be classified into two categories:

- Spectral methods [11]
- Spatial methods [12]

In this paper, we utilize the widely used convolution operator [13], which can be regarded as both the spectral operator and spatial operator.

Graph convolutions are applied to graph networks in the non-probabilistic GAE [4] and VGAE [4] architectures.

GAE firstly transforms each node into latent representation (i.e., embedding) via GCN and then aims to reconstruct some part of the input. GAEs proposed in [4], [14], and [15] intend to

reconstruct the adjacency via decoder while GAEs developed in [16] attempt to reconstruct the content.

Variational Graph Autoencoders (VGAE) propose similar to GAE approach but with some differences. The difference between VGAEs and GAEs is that VGAE embeds the input to a *distribution* rather than a point, and decoder produces an output using a variational approximation [17]. Such architecture allows Variational Auto encoders to generate new data from the original source dataset. At the same time, regular autoencoders only produce output similar to the input [18].

While VGAE is a framework, there are different variations and implementations of it. For example, an attributed network embedding model using VGAE is proposed [19] for learning both node and attribute representations in the same space.

Also, there is a variance of VGAE, which was created to learn rating embeddings by considering them for users and review texts [20].

It is also worth noting some legal domain studies related to the usage of DNNs and GCNs to investigate legal datasets. Text-guided Graph Reasoning approach [21] was introduced for combining text representation and structure knowledge. This approach solves graph completion tasks and utilizes R-GCN and GAT networks. It is also model agnostic and can be implemented in other GNNs. Another approach for working with legal graphs is to use a DNN hybrid model [22], which extracts events in the knowledge map. In addition, this approach combines the advantages of convergence and iterative DNNs for extracting events for common convergence and bidirectional iterative DNNs. In a fuzzy DNN approach, input data is converted into a double precision variable [23]. With such an approach, each character sequence is forcibly transformed into integer variables, which are transformed into floating-point double precision variables.

## 3. Methodology

### 3.1. Dataset

For this work, the hierarchical set of Ukrainian legal acts adopted by the Ukrainian parliament (Verkhovna Rada of Ukraine) and Ukrainian government (the Cabinet of Ministers of Ukraine) was used to prepare the customized dataset with the structure shown in Table 1. The representative part of the dataset can be accessed on Kaggle open data platform [24].

Some of the features in Table 1 have the following additional characteristics:

- **Status (one of the values):** undefined (0), taking effect (2), renewed (4), effective (5). “undefined” indicates the missing data about the status of the current act.
- **Types:** Law (1), Decree (20), Order (30), Codified Law (124), Agenda (201), Constitution (100), duplicated value for Constitution (216). It is possible for the act to contain more than one type; for instance, codified laws (124) are just laws (1) too.
- **Institutions:** The set of state institutions IDs, which passed the law.

Also, the dataset contains some relations data between the legal acts in the hierarchical structure that are described in Table 2.

**Table 1**

The structure of the customized dataset for the hierarchical set of Ukrainian legal acts

Feature	Type	Example
Document ID	ID	12
Title	ASCII text	"Constitution of Ukraine"
Status	Integer Enum	2
Types	Array of integer Enums	124 1
Institutions	Array of IDs	123 15 7
Text content	ASCII text	"This law regulates ..."

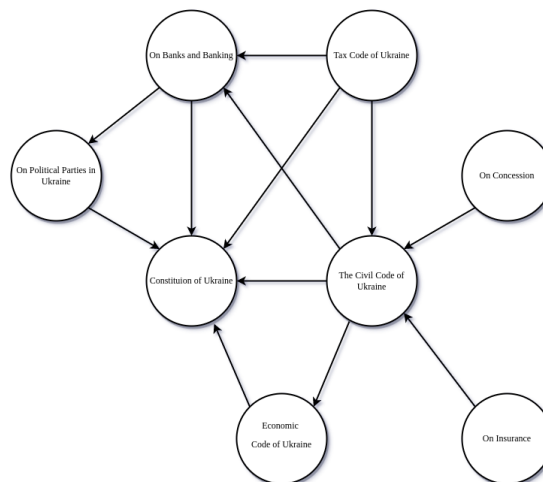
**Table 2**

Relation between the legal acts (in Table 1)

Feature	Type	Example
Source document	ID	124
Target document	ID	305
Relation type	Integer Enum	2

**Relation type (one of the values):** Origin/root (2), Relates to (6)

Legal acts and, more specifically, links between them are represented with a directed graph, in which acts are nodes and relations are represented through edges.

**Figure 1:** Example of dataset structure.

However, the graph is incomplete, and there are cliques, which are not connected to each other, which results in having a certain amount of sub-graphs.

Here and later, such attributes are used to evaluate data:

- **Number of edges**
- **Depth:** Median and mean depth of each node in graph
- **Maximum clique size**
- **Degree centrality:** Some legal acts have more connections, than others (i.e: Constitution of Ukraine, Codified Laws)
- **Eigenvector centrality:** Another metric for node centrality. The difference with regular degree centrality is that Eigenvector centrality measures a node’s importance while considering the importance of its neighbors.

**Table 3**  
Dataset characteristics

Characteristic	Value
Number of graphs	427
Number of nodes	68841
Number of edges	89916
Mean depth	5.95
Median depth	11
Maximal clique size	8
Mean degree centrality	6.7302e-05
Median degree centrality	5.1527e-05
Mean eigen centrality	0.00029
Median eigen centrality	2.1227e-15

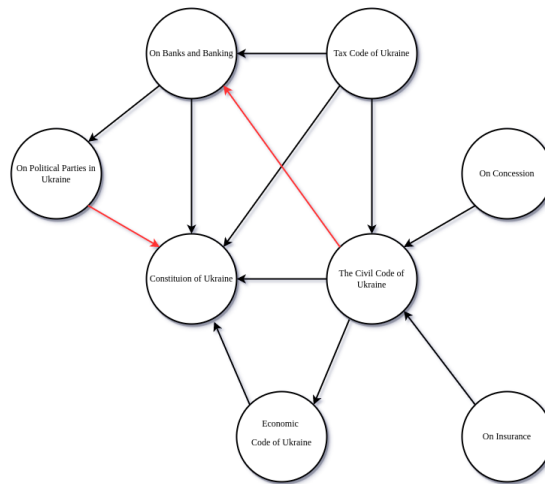
### 3.2. Workflow

In order to construct GNNs, non-categorical strings (acts titles and plain text contents) needed to be transformed into tensor features. It was done by utilizing the sentence-transformer all-MiniLM-L6-v2 model based on MiniLM model [25], which maps sentences and paragraphs to a 384 dimensional dense vector space.

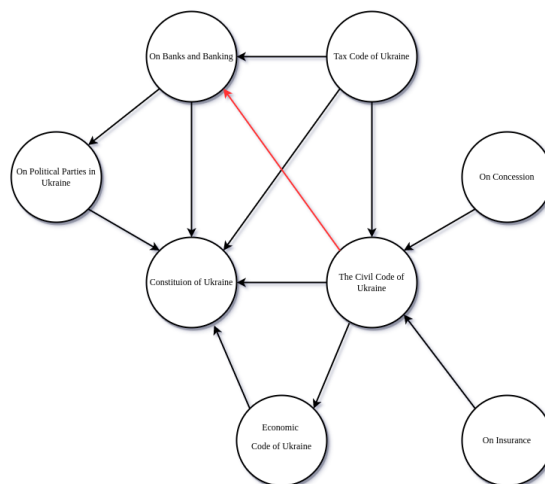
After transforming the dataset to the graph structure, it was split into 3 subsets - train (Table 4), validation (Table 5), and test (Table 6). This split was done by randomly splitting edges of the graph and was performed such that the train split does not include some edges that were present in the validation and test splits (see explanations in Figure 2 and Figure 3).

In the same way, the validation split does not include some edges that were present in the test split (see explanations in Figure 3). It is also worth noting that the set of nodes and node features were the same in all the sets because the model was used to predict links only.

To ensure that all the sets are represented equally, they were evaluated with regard to the same characteristics as the whole base dataset.



**Figure 2:** Structure of train set. The red arrows represent message edges which are included in validation and test subsets but are not included in the train subset.



**Figure 3:** Structure of validation set. The red arrows represent message edges that are included in the test subset but not in the validation subset.

### 3.3. Exploratory Data Analysis

As one can see from applying Exploratory Data Analysis (EDA) to train, validation, and test subsets, they have an equal number of nodes but a different number of edges. Also, one can see some differences in mean and median values of depth, degree centrality, and eigenvector centrality (Tables 4-6). To investigate this issue, distribution charts were constructed for each of these characteristics: depth (Figure 4), degree centrality (Figure 5) and eigenvector centrality

**Table 4**  
Train subset characteristics

Characteristic	Value
Number of graphs	466
Number of nodes	68841
Number of edges	56648
Mean depth	4.9106
Median depth	11
Maximal clique size	6
Mean degree centrality	5.2208e-05
Median degree centrality	3.1722e-05
Mean eigen centrality	0.0002
Median eigen centrality	2.4598e-14

**Table 5**  
Validation subset characteristics

Characteristic	Value
Number of graphs	467
Number of nodes	68 841
Number of edges	62942
Mean depth	4.7862
Median depth	10
Maximal clique size	6
Mean degree centrality	5.4953e-05
Median degree centrality	6.0103e-05
Mean eigen centrality	0.0002
Median eigen centrality	6.4074e-14

(Figure 6) distributions. While mean and median values may differ, the overall distributions are very similar across all subsets. Validation and test subsets can be considered representative ones with regard to the train subset.

It is also worth mentioning, that edges represented in Fig.2 and Fig.3 are used only for message passing. This is done to exchange neighborhood information and enhance node representations. Edge labels and edge label indices are completely isolated and are not shared between sets. Usually, Graph Autoencoders use the same edges for message passing and train in validation sets, but here we enforce additional isolation by additionally removing message passing edges from the training subset to prevent possible data leaks.

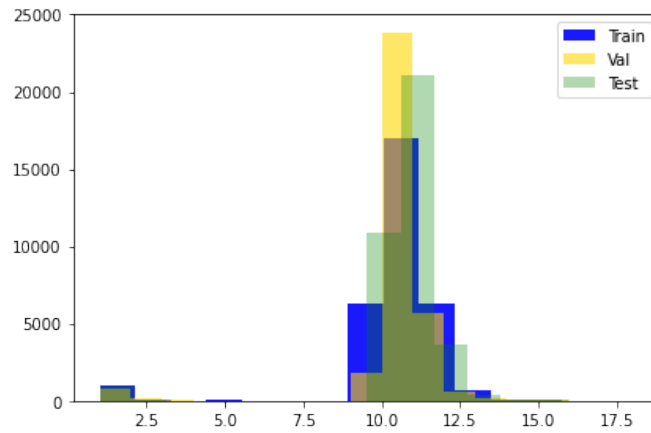
### 3.4. Models

Four different models were created to investigate the effect of GAE/VGAE for efficient link predictions in the customized dataset. All models implement GAE architecture, consisting of an encoder and a decoder. The encoder takes data from input and transforms it into a lower dimensional embedding. Then the decoder takes this lower dimensional embedding and

**Table 6**

Test subset characteristics

Characteristic	Value
Number of graphs	437
Number of nodes	68841
Number of edges	80925
Mean depth	5.7279
Median depth	11
Maximal clique size	8
Mean degree centrality	6.3060e-05
Median degree centrality	5.3643e-05
Mean eigen centrality	0.0002
Median eigen centrality	6.4101e-15

**Figure 4:** Depth distribution across sets

reconstructs the original input [26]. In GAE architecture, the loss function determines the amount of information lost during decoding.

In this experiment, the following GCN models were used:

- GAE with two layer GCN encoder (GCN) shown in Fig. 7,
- Single layer Linear GCN encoder (LGCN) shown in Fig. 8,
- Variational two-layer GCN encoder (VGCN) shown in Fig. 9,
- Variational Single layer Linear Graph Convolutional network (VLGCN) shown in Fig. 10.

All models implement GCN encoder and dot product decoder. In VGAE-based models, encoders output mean and variance vectors, which are converted to z-embedding. Encoders of regular GAE models output z-embedding directly.

The training was performed with a calculation of some standard metrics (like area under the curve (AUC), precision, recall, mean squared error, r2 score, f1) after 1000 epochs for training, validation, and test subsets. Each model's training process was performed 10 times to calculate



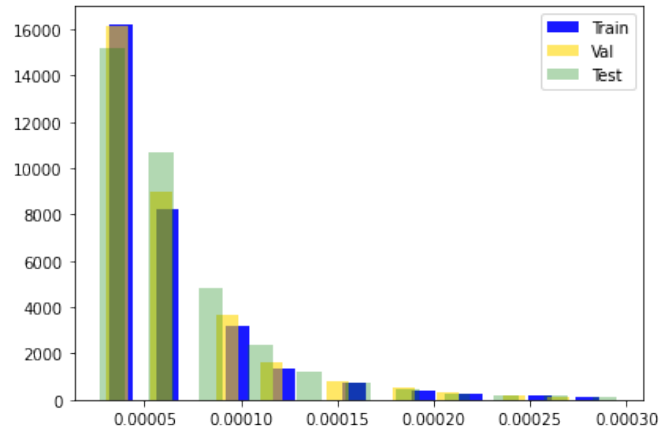


Figure 5: Degree centrality distribution across sets

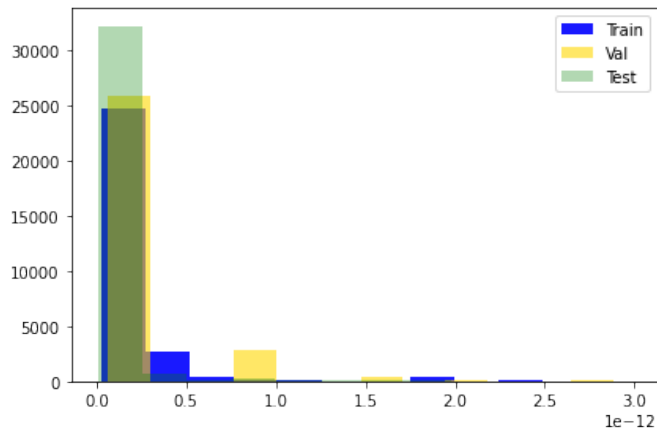


Figure 6: Eigenvector centrality distribution across sets

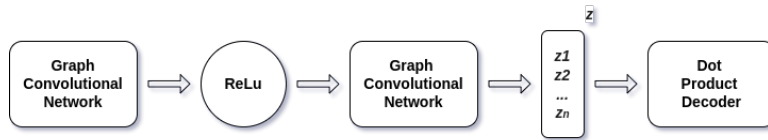


Figure 7: GCN model schema

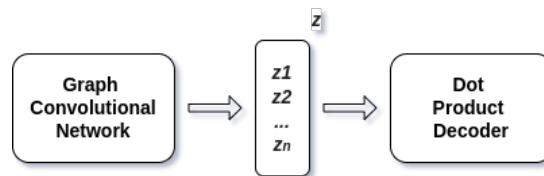


Figure 8: LGCN model schema

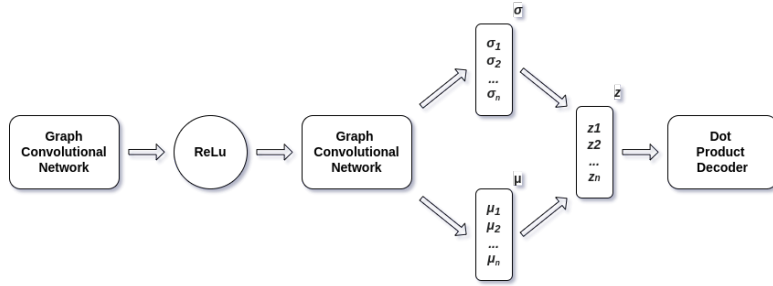


Figure 9: VGCN model schema

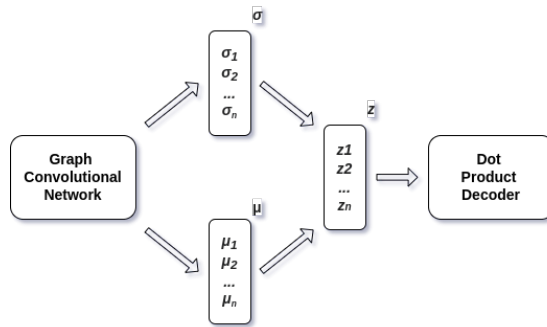


Figure 10: VLGCN model schema

mean performance and standard deviation. Each training iteration was done with a randomly generated seed, and models were reset each time.

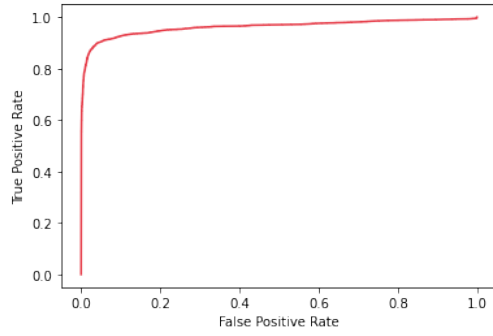
## 4. Experimental

As one can see in Table 7, all models demonstrate relatively high performance metrics. However, model performance is pretty similar, and the difference between AUC values in the worst and best performing models is lower than 2%. As one can observe in Figure 15, ROC curves of all models almost overlap each other.

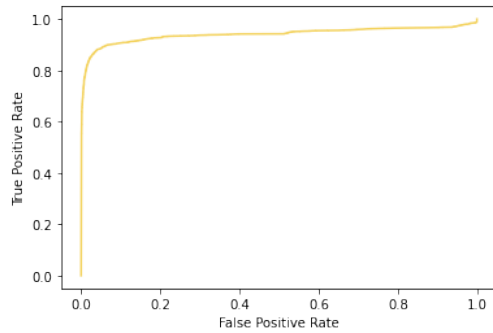
Table 7

Test metrics for researched models

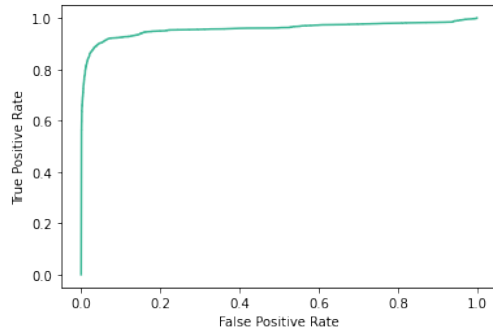
Metric	GCN	LGCN	VGCN	VLGCN
AUC	<b>0.9623 ± 0.0005</b>	0.9416 ± 0.0002	0.9585 ± 0.0015	0.9515 ± 0.0003
MSE	<b>0.1546 ± 0.0002</b>	0.1636 ± 3.0397e-05	0.1625 ± 0.0005	0.1796 ± 0.0015
R2	<b>0.3814 ± 0.0011</b>	0.3452 ± 0.0001	0.3499 ± 0.0021	0.2812 ± 0.0062
Precision	<b>0.9730 ± 0.0003</b>	0.9631 ± 0.0001	0.9718 ± 0.0008	0.9719 ± 0.0001
Recall	<b>0.9962 ± 0.0004</b>	0.9813 ± 5.4487e-05	0.9942 ± 0.0004	0.9762 ± 5.5481e-05
F1	<b>0.9845 ± 0.0003</b>	0.9721 ± 6.1622e-05	0.9828 ± 0.0005	0.9740 ± 9.7375e-05



**Figure 11:** ROC curve for GCN model



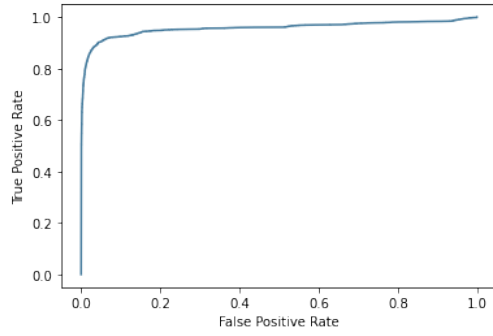
**Figure 12:** ROC curve for LGCN model



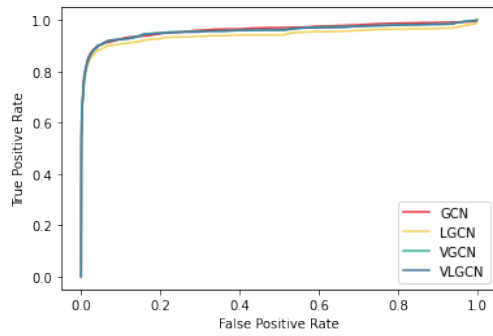
**Figure 13:** ROC curve for VGCN mode

Such a high performance may be explained by the fact that usually, the legal acts references consist of mentioning them in the text content part of other acts. Thus, the model is mostly trained to find embeddings of the titles of the legal acts in the legal acts text content embedding.

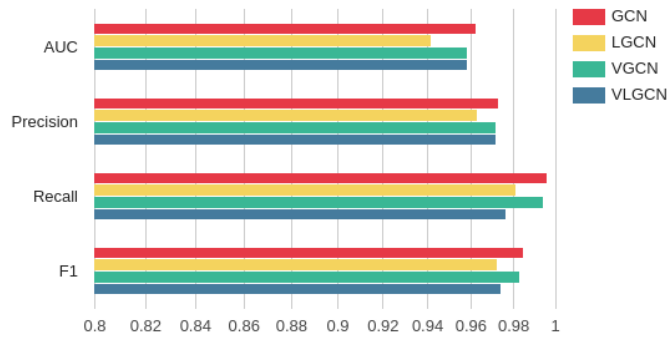
It should be noted that VGCN performed worse than the traditional GCN model, even though the VLGCN model performed better than the LGCN model. However, by most metrics, traditional GCN has the best values among all of the 4 models.



**Figure 14:** ROC curve for VLGCN model



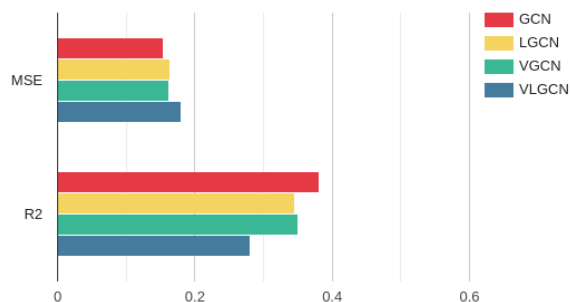
**Figure 15:** ROC of all of evaluated models



**Figure 16:** Combined model AUC, Precision, Recall and F1 metrics

The reason for poorer VGAE model behavior may be that relations between acts are strict and direct, and any augmentation of existing data or generating new data may actually reduce model performance.

Also, for this specific dataset, the amount of GCN layers seems to be more important than differences between GAE and VGAE architectures, which is especially clearly seen in Fig 16 where LGCN performs worse than GCN and VLGCN performs worse than VGCN.



**Figure 17:** Combined model MSE and R2 metrics

## 5. Discussion

Another popular solution to link prediction task in GNNs is LightGCN model [27]. LightGCN model implements GCN by using only neighborhood aggregation for collaborative filtering. It is done by linearly propagating nodes embeddings on interaction graph [27].

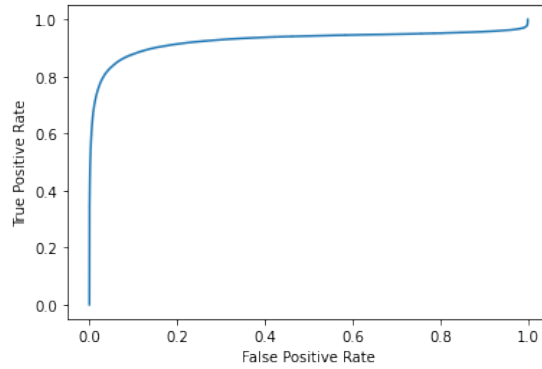
To further investigate GAE and VGAE performance, the LightGCN model was trained on the same dataset and evaluated with the same metrics, which are shown in Table 8. Also, the ROC curve was constructed, as shown in Fig 18. Considering all LightGCN metrics, we can conclude that GAE/VGAE architecture is better suited for the link prediction task.

**Table 8**  
LightGCN and GCN metrics comparison

Metric	LightGCN	GCN
AUC	0.9253	0.9623
MSE	0.1840	0.1546
R2	0.2788	0.3814
Precision	0.9405	0.9730
Recall	0.9466	0.9962
F1	0.9486	0.9845

The following observations were made after these experiments. All models have quite similar performance; however, the GCN model (the one with two GCN layers) performs better than other models.

Moreover, models with singular GCN (LGCN and VLGCN) perform observably worse than models with 2 GCNs (GCN and VGCN). The difference between GAE and VGAE based models is insignificant; however, variational models perform slightly worse. The cause of such an effect may be the nature of the specific dataset, upon which these models were evaluated. This comes to the conclusion that the amount of GCNs may be more significant in some cases than the



**Figure 18:** ROC curve for LightGCN model

differences between GAE and VGAE architectures.

By experimenting with the configuration of GCN layers, prediction performance may be further improved. For example, the other further improvements of the proposed approach can be implemented due to hybridization of graph, convolutional, variational, and other network components that was verified in our previous works [28, 29, 30] and other researches [31, 32].

Another possible research subject may include training models on a dataset with act content split into different parts (for example, by articles) instead of training by the complete act text content.

Another promising research topic may be studying GAE/VGAE models behaviour on knowledge graphs constructed on the legal acts dataset. Such graph may include information about institutions, date of publication, and even data about parliament members.

Also, the aforementioned models were compared with the LightGCN model, which was trained on the same dataset. Comparing LightGCN model evaluation metrics with GAE / VGAE leads to the conclusion that the latter produces more competitive results.

## 6. Conclusions

Finally, in the context of many practical fields, the link prediction problem is fundamental to better understanding links between objects in hierarchical constructions such as networks. To solve it, various graph-based DNNs, like GAE and VGAE, have become powerful tools. In this work, some variants of graph-based DNNs have been demonstrated to solve the link prediction problem among text objects, specifically, legal acts. The correspondent customized dataset in the form of a hierarchical set of Ukrainian legislation documents (based on the legal acts adopted by the Ukrainian parliament and the Government of Ukraine). The exploratory data analysis (EDA) was performed, and the structure of training, validation, and test subsets was analyzed. Several variants of GAE and VGAE models were proposed and applied to the dataset, namely, single layer Linear GCN encoder (LGCN), variational two-layer GCN encoder (VGCN), and variational LGCN (VLGCN). The comparative analysis of all considered models was performed based on several standard metrics. The conclusion was drawn that all models have quite similar

performance; however, the GCN model (the one with two GCN layers) performs better than other models. The LightGCN model, in comparison with GAE/VGAE, leads to the conclusion that the latter has the higher performance metrics. The models with singular GCN (LGCN and VLGCN) perform observably worse than models with 2 GCNs (GCN and VGCN). In the context of legal act hierarchy, prediction performance may be further improved by considering other information about institutions, date of publication, and even data about parliament members. In general, other further improvements of the proposed method are possible for other hybridization variants of graph, convolutional, variational, and other network components.

## Acknowledgments

The work was partially supported by “Knowledge At the Tip of Your fingers: Clinical Knowledge for Humanity” (KATY) project funded from the European Union’s Horizon 2020 research and innovation program under grant agreement No. 101017453.

## References

- [1] L. Getoor, C. P. Diehl, Link mining: a survey, *Acm Sigkdd Explorations Newsletter* 7 (2005) 3–12.
- [2] L. Lü, T. Zhou, Link prediction in complex networks: A survey, *Physica A: statistical mechanics and its applications* 390 (2011) 1150–1170.
- [3] S. Kerrache, R. Alharbi, H. Benhidour, A scalable similarity-popularity link prediction method, *Scientific Reports* 10 (2020) 6394. doi:10.1038/s41598-020-62636-1.
- [4] T. N. Kipf, M. Welling, Variational graph auto-encoders, 2016. URL: <https://arxiv.org/abs/1611.07308>. doi:10.48550/ARXIV.1611.07308.
- [5] S. Zhang, H. Tong, J. Xu, R. Maciejewski, Graph convolutional networks: a comprehensive review, *Computational Social Networks* 6 (2019). doi:10.1186/s40649-019-0069-y.
- [6] Z. Ye, Y. J. Kumar, G. O. Sing, F. Song, J. Wang, A comprehensive survey of graph neural networks for knowledge graphs, *IEEE Access* 10 (2022) 75729–75741. doi:10.1109/ACCESS.2022.3191784.
- [7] E. Filtz, Knowledge Graphs for Analyzing and Searching Legal Data, Ph.D. thesis, Vienna University of Economics and Business, 2021.
- [8] V. Singh, P. Lio, Towards probabilistic generative models harnessing graph neural networks for disease-gene prediction, *arXiv preprint arXiv:1907.05628* (2019).
- [9] O. Alienin, O. Rokovyi, Y. Gordienko, Y. Kochura, V. Taran, S. Stirenko, Artificial intelligence platform for distant computer-aided detection (cade) and computer-aided diagnosis (cadx) of human diseases, in: *The International Conference on Artificial Intelligence and Logistics Engineering*, Springer, 2022, pp. 91–100.
- [10] Y. Yakimenko, S. Stirenko, D. Koroliouk, Y. Gordienko, F. M. Zanzotto, Implementation of personalized medicine by artificial intelligence platform, in: *2nd International Conference on Soft Computing for Security Applications*, Springer, 2022.
- [11] M. Niepert, M. Ahmed, K. Kutzkov, Learning convolutional neural networks for graphs, in: M. F. Balcan, K. Q. Weinberger (Eds.), *Proceedings of The 33rd International Conference*

- on Machine Learning, volume 48 of *Proceedings of Machine Learning Research*, PMLR, New York, New York, USA, 2016, pp. 2014–2023. URL: <https://proceedings.mlr.press/v48/niepert16.html>.
- [12] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and locally connected networks on graphs, 2013. URL: <https://arxiv.org/abs/1312.6203>. doi:10.48550/ARXIV.1312.6203.
- [13] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, CoRR abs/1609.02907 (2016). URL: <http://arxiv.org/abs/1609.02907>. arXiv:1609.02907.
- [14] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, C. Zhang, Attributed graph clustering: A deep attentional embedding approach, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 3670–3676. URL: <https://doi.org/10.24963/ijcai.2019/509>. doi:10.24963/ijcai.2019/509.
- [15] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, C. Zhang, Adversarially regularized graph autoencoder for graph embedding, 2018. URL: <https://arxiv.org/abs/1802.04407>. doi:10.48550/ARXIV.1802.04407.
- [16] C. Wang, S. Pan, G. Long, X. Zhu, J. Jiang, Mgae: Marginalized graph autoencoder for graph clustering, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 889–898. URL: <https://doi.org/10.1145/3132847.3132967>. doi:10.1145/3132847.3132967.
- [17] I. Gatopoulos, J. M. Tomczak, Self-supervised variational auto-encoders, Entropy 23 (2021) 747. URL: <https://doi.org/10.3390/e23060747>. doi:10.3390/e23060747.
- [18] W. Yu, G. Zeng, P. Luo, F. Zhuang, Q. He, Z. Shi, Embedding with autoencoder regularization, in: H. Blockeel, K. Kersting, S. Nijssen, F. Železný (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 208–223.
- [19] Z. Meng, S. Liang, H. Bao, X. Zhang, Co-embedding attributed networks, 2019, pp. 393–401. doi:10.1145/3289600.3291015.
- [20] X. Li, J. She, Collaborative variational autoencoder for recommender systems, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 305–314. URL: <https://doi.org/10.1145/3097983.3098077>. doi:10.1145/3097983.3098077.
- [21] L. Li, Z. Bi, H. Ye, S. Deng, H. Chen, H. Tou, Text-guided legal knowledge graph reasoning, in: B. Qin, Z. Jin, H. Wang, J. Pan, Y. Liu, B. An (Eds.), Knowledge Graph and Semantic Computing: Knowledge Graph Empowers New Infrastructure Construction, Springer Singapore, Singapore, 2021, pp. 27–39.
- [22] L. Zhou, Event scene method of legal domain knowledge map based on neural network hybrid model, Applied Bionics and Biomechanics 2022 (2022) 5880595. URL: <https://doi.org/10.1155/2022/5880595>. doi:10.1155/2022/5880595.
- [23] Y. Xie, Application of deep neural network algorithm in the analysis of legal precedent citation basis, Mobile Information Systems 2022 (2022) 3383428. URL: <https://doi.org/10.1155/2022/3383428>. doi:10.1155/2022/3383428.



- [24] Ukrainian legal acts dataset, 2022. URL: <https://www.kaggle.com/datasets/vladyslavshlianin/ukrainian-legal-acts>, accessed on Sep, 24, 2022.
- [25] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, M. Zhou, Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), *Advances in Neural Information Processing Systems*, volume 33, Curran Associates, Inc., 2020, pp. 5776–5788. URL: <https://proceedings.neurips.cc/paper/2020/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.
- [26] W. Wang, Y. Huang, Y. Wang, L. Wang, Generalized autoencoder: A neural network framework for dimensionality reduction, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2014.
- [27] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: Simplifying and powering graph convolution network for recommendation, in: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20*, Association for Computing Machinery, New York, NY, USA, 2020, p. 639–648. URL: <https://doi.org/10.1145/3397271.3401063>. doi:10.1145/3397271.3401063.
- [28] Y. Gordienko, K. Kostiukevych, N. Gordienko, O. Rokovyi, O. Alienin, S. Stirenko, Deep learning with noise data augmentation and detrended fluctuation analysis for physical action classification by brain-computer interface, in: *2021 8th International Conference on Soft Computing & Machine Intelligence (ISCMCI)*, IEEE, 2021, pp. 176–180.
- [29] K. Kostiukevych, Y. Gordienko, N. Gordienko, O. Rokovyi, O. Alienin, S. Stirenko, Convolutional and recurrent neural networks for physical action forecasting by brain-computer interface, in: *11th IEEE Int. Conf. on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, IEEE, 2021.
- [30] K. Kostiukevych, Y. Gordienko, N. Gordienko, O. Rokovyi, S. Stirenko, Hierarchy of hybrid deep neural networks for physical action classification by brain-computer interface, in: *Modern Machine Learning Technologies and Data Science*, CEUR, 2022.
- [31] S. Zhang, H. Tong, J. Xu, R. Maciejewski, Graph convolutional networks: Algorithms, applications and open challenges, in: *International Conference on Computational Social Networks*, Springer, 2018, pp. 79–91.
- [32] F. Yang, H. Zhang, S. Tao, Hybrid deep graph convolutional networks, *International Journal of Machine Learning and Cybernetics* (2022) 1–17.