Analyzing the Citation Rate of a Scientific Paper Using ML and Graph Algorithms

Vladimir Zakharov, Sergey Fedchin, Artem Chebykin

Fall 2024

Contents

1 Introduction **2**

1.1 Paper Overview 2

1.2 Field Overview 2

2 Data Acquisition and Preprocessing **3**

3 Exploratory Data Analysis **3**

3.1 General Properties 4

3.2 Degree Distribution 4

3.3 Centralities 5

3.4 Time Properties 5

3.5 Clusterization 6

4 Methodology **8**

4.1 Problem statement 8

4.2 Baseline 8

4.3 Our solution 10

5 Conclusion **11**

References **12**

1 Introduction

1.1 Paper Overview

Citation rate is an important metric to know about a scientific paper. It is often referred to as a KPI for researchers. Governmental agencies and scientists pay attention to the citation rate, when evaluating the quality of the paper. Hence, it is important to understand what exactly affects the citation of your work. In our research, we investigate the connection between various properties of scientific papers and their citation rates, using machine learning and graph algorithms.

1.2 Field Overview

Due to the importance of citation rate prediction, there are quite a few papers on the subject, which implement various decisions of the stated problem. Those methods range from the most simple ones, which analyze the abstracts of papers with using just basic word embeddings; to a more complex, which take into account the structure of the citation graph. We will be using those simple models as a baseline for our own solutions, which will incorporate graph neural networks(GNN) in combination with the use of abstracts analysis. The studies, we have analyzed are:

There are quite some works, that focused primarily on the graph features of the network,like this [ZW24], which disregarded any meta-information about the paper, such as authors, scientific journal, which posted it, or even the abstract

This work [FDG20] on the other hand, took into account the semantic structure of papers. However, this study used graph information only to create vertex features (betweenness, degree, etc.), and did not implement any graph ML approaches to perform the task

There is also a number of papers, which tried to use various graph algorithms(like PageRank) to predict citation rates [Gao+24]. Those methods showed better results, than any baseline, but it ignored any additional information about the paper nonetheless

If we want to talk about the use of papers abstracts, then[Rom+21] is the work to relate. In this study, a multitude of different models were used to create text embeddings for the papers, which were then used to predict citation rate by the paper abstract.

The next paper inspired the way we interpreted the task, using older citation to predict the future ones[Jr+24]. The idea of splitting given graph by time, and using older and newer papers differently, as well as the use of a couple rather interesting ML algorithms, allowed authors to reach quite impressive results in terms of citation rate prediction .

2 Data Acquisition and Preprocessing

The amount of data and its quality is the basis for any data science research and there are quite a number of different datasets in this area. To explore the dependencies between the paper citation rate and its inner properties, we have chosen the dataset: HepTh dataset .The dataset represents the scientific paper citation graph in the field of high-energy physics in the years 1991 to 2003. There were 2 main reasons, for why this dataset was selected. Firstly, almost no research had been conducted on this data, which allowed us to implement baseline decisions from various other papers, and see how they perform on a different dataset. And secondly, due to limited resources, we did not want to analyze extremely large dataset, so we needed something a touch smaller than that.

Couple of words about the structure of the data. An oriented edge between vertices a→b shows, that paper a cites paper b. Apart from the graph information itself, we are also provided with various features of each paper, which are, however, not standardized. Some features only appear in few papers, while others are present for each. We have decided to keep and work with the following features present for each paper: date of publication, authors, title + abstract.

The task of processing the dates was particularly tricky one since that part of the data was stored in a multitude of different formats, and they had to be standardized. (We have chosen yyyy-mm-dd format for it). Furthermore, the graph included some edges that led from an older paper to a more recent one, which didn't make sense; hence, those edges were removed from our data. We have also transformed the list of authors from a string into an actual Python list and parsed the abstract from the metadata files provided in the dataset.

After all the described manipulations, the data was ready for further exploration and analysis.

3 Exploratory Data Analysis

All calculations with the graph were performed using NetworkX library in Python because it has a well-written documentation and provides a convenient interface.

3.1 General Properties

Let V be the set of all papers (nodes) and $E = \{(v_1, v_2) : v_1, v_2 \in V, v_1 \text{ cites } v_2\}$ all the citations between papers (directed edges). Then $G(V, E)$ is our directed citation graph. For this graph

$$|V| = 27.376$$

$$|E| = 351.025$$

The original graph had more nodes – 27.7K, but we discovered that it had around 140 weakly connected components (connected components in graph G' obtained from G by disregarding the edges orientation). Most of them were of sizes 3-5 so we decided to leave only the main component and delete all the stray nodes. Due to the graph's nature, it cannot be strongly connected (and number of strongly connected components is exactly $|V|$), since strong connectivity means that there exists a path from any node v to any other node u in the component, but it yields a cycle, which is impossible in G, as stated earlier.

Some graph properties are defined for directed graphs only if they are strongly connected: diameter — length of the "longest shortest path", average path length, girth – length of the smallest cycle (usually only defined for undirected graphs). What we can calculate is the density of G:

$$D(G) = \frac{|E|}{2\binom{|V|}{2}} \approx 0.00046,$$

so the graph is rather sparse, which makes sense if we take into account the nature of G.

One other characteristic that we can compute is the average clustering coefficient, which is a measure of the degree to which nodes in a graph tend to cluster together.

$$\bar{C}(G) = \frac{1}{n} \sum_{v \in V} C_v \approx 0.156,$$

where C_v is the local clustering coefficient of the node v, defined as

$$C_v = \frac{|E(G(N_v))|}{\binom{V(G(N_v))}{2}},$$

if $G(N_v)$ is the subgraph of G with only the immediate neighbours of v as nodes.

3.2 Degree Distribution

One of the important aspects of any graph is the distribution of the degrees of nodes. Since our graph is directed, we will have 2 distributions: one for the in-degree (number of incoming edges) and one for the out-degree (number of outgoing edges). Both of their distributions in log-scale are show on Figure 1.

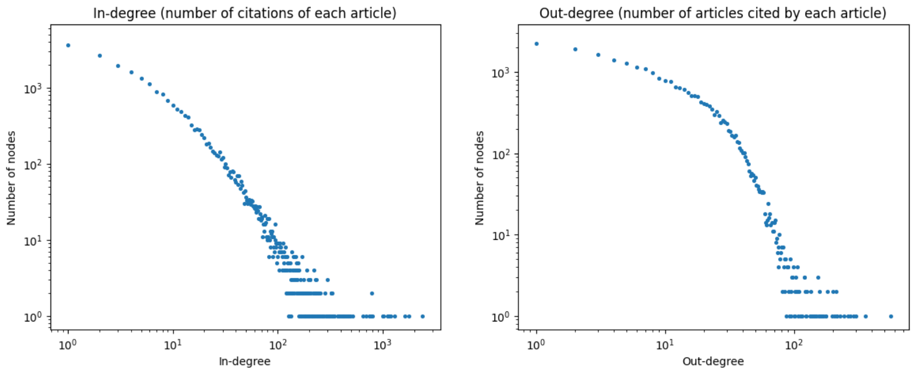

Figure 1: Degrees distribution.

The first distribution is rather typical for a scale-free network and follows the power law. However, it is not exactly the case with the out-degree distribution. The possible explanation to this is the following: if we look at the graph as a temporal graph (we have dates of each paper’s publishing) out-degree of each node is determined on creation and does not change over time. In-degree, on the other hand, has a behaviour that is much closer to how degrees act in a usual social graph, developing over time.

3.3 Centralities

Centralities are characteristics of nodes that determine how ”important” or ”popular” each of them is. There are several types of them and each measures this ”importance” in its own way. We will visualize distribution of three centralities.

a) Eigenvector centrality. In general, relative scores are assigned to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes.

b) Closeness centrality. It is calculated as the reciprocal of the sum of the length of the shortest paths between the node and all other nodes in the graph. It measures how ”close” the node is to the rest of the graph.

c) Betweenness centrality. The betweenness centrality for each node is the number of shortest paths that pass through this node. In other words, it determines whether the node is a hub.

3.4 Time Properties

Since our graph can be interpreted as a temporal graph, we can study the way its properties change over time. For instance, we discovered that the

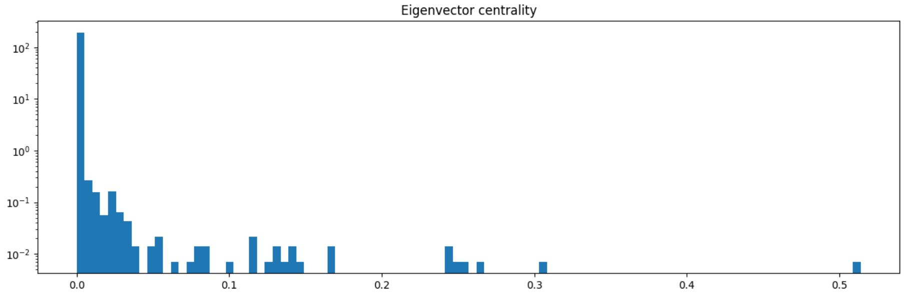

Figure 2: Eigenvector centrality distribution.

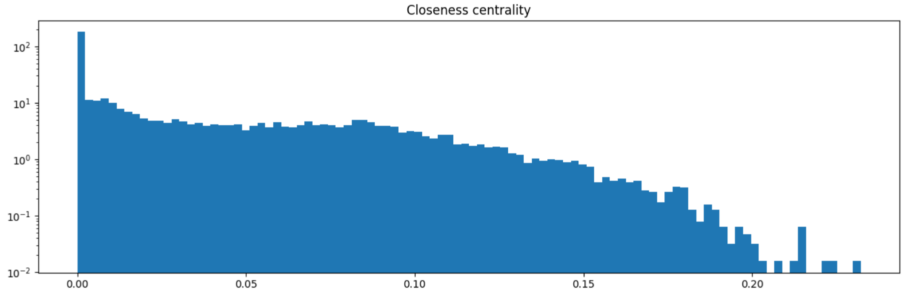

Figure 3: Closeness centrality distribution.

number of new papers per unit of time increases but the rate of the increase decreases. See Figure 5 for the plots.

Another interesting thing that we found is how average in-degrees and out-degrees change (see Figure 6). A downwards trend in the end for in-degree and upwards trend in the beginning are noticeable. Explanation to this phenomena is rather simple: older papers had more time to gain popularity and therefore have more incoming edges, whereas papers from 2003 do not have any papers in our dataset that cite them. The opposite happens with the out-degree. Earlier papers cited even earlier papers (from 1980s) that were not included in our graph and thus have little to no outgoing edges. It is also worth noticing that the growth of out-degree slows down closer to 2003. It means that almost enough papers are included in the graph at that time to publish a paper sourcing only these modern papers in our dataset.

3.5 Clusterization

Due to the duality of our problem, clusterization can be done either using node features (mainly papers' abstracts) or considering the graph structure (which is usually called community detection). As a measure of quality of the partition P of graph G we have chosen the metric modularity, which for directed graphs

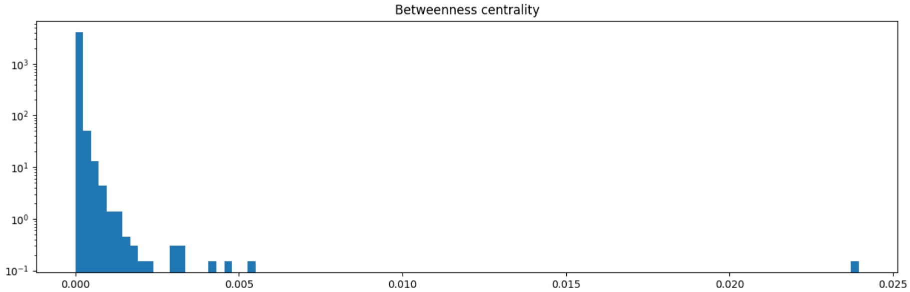

Figure 4: Eigenvector centrality distribution.

is defined as

$$\textbf{Modularity}(G,P)=\frac{1}{|E|}\sum_{u,v\in V}\left[\left(A_{u,v}-\frac{d_{\text{out}}(u)\cdot d_{\text{in}}(v)}{|E|}\right)\cdot\delta_P(u,v)\right],$$

where A is the adjacency matrix and $\delta_P(u,v)$ is defined as

$$\delta_P(u,v)=\begin{cases}1, & \text{if nodes } u \text{ and } v \text{ lie in the same community in partition } P\\ 0, & \text{otherwise}\end{cases}.$$

First we did community detection on the graph. We compared Louvain and label propagation community detection algorithms since most other methods were extremely slow and thus not suitable for our graph. The best modularity was achieved by Louvain algorithm with partition into 31 classes with fixed seed compared to 969 classes using label propagation.

To do clusterization we have divided the articles into the same number of clusters according to their abstracts using text embedding from ‘glove-wiki-gigaword-50‘ model (we averaged the embeddings of each word in the abstract) and KMeans algorithm. We also tried agglomerative clustering and Gaussian mixture. Modularity for all of them was really poor, which should not come as a surprise. Visualizations of resulting clustering can be found on Figures 7 and 8. The comparison of clustering algorithms can be seen in Table 1.

Algorithm	Number of clusters	Modularity
Louvain	31	0.655
Label Propagation	969	0.541
Agglomerative Clustering	31	-0.0002
Gaussian Mixture Model	31	0.0008

Table 1: Comparison of different clusterization algorithms.

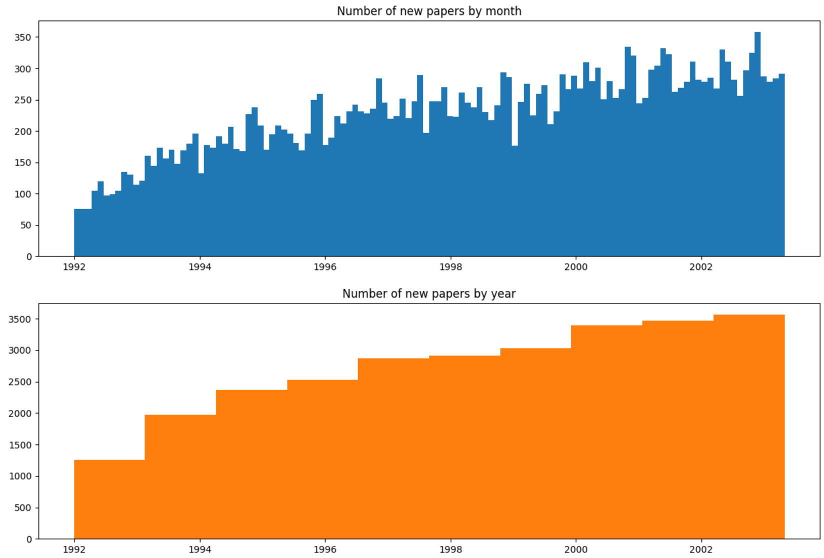

Figure 5: Number of new papers published each two months and each year.

4 Methodology

4.1 Problem statement

Our main goal was to predict citation rates of the papers. We initially thought of solving a link prediction task (predicting the existence of edge between two nodes) since it seemed logical enough for our task. However, upon further considerations we eventually switched to node classifications. We classified citation rates into 4 classes using quartiles of in-degrees (this manner was inspired by scientific journals' tiers). This approach gave us even-ish distribution of classes. Note that the class boundaries are not evenly distributed, since our graph is a scale-free network.

To form target values and split our graph into train and test samples the following approach was used. Split the graph into two parts: all the nodes with publish date before 2001-01-01 (the date choice was arbitrary) and the rest of them, which will be referred to as old nodes and new nodes respectively. The new nodes have some edges into the old ones (new papers citing old papers), the number of which (converted into a class) for each old node was used as its target value. Note that the edges between old nodes were not considered in the target. By splitting old nodes into train and test sets we got our final data.

4.2 Baseline

Before implementing any baseline we studied how other papers approached this problem. The state-of-the-art approach for baseline solutions, is to use

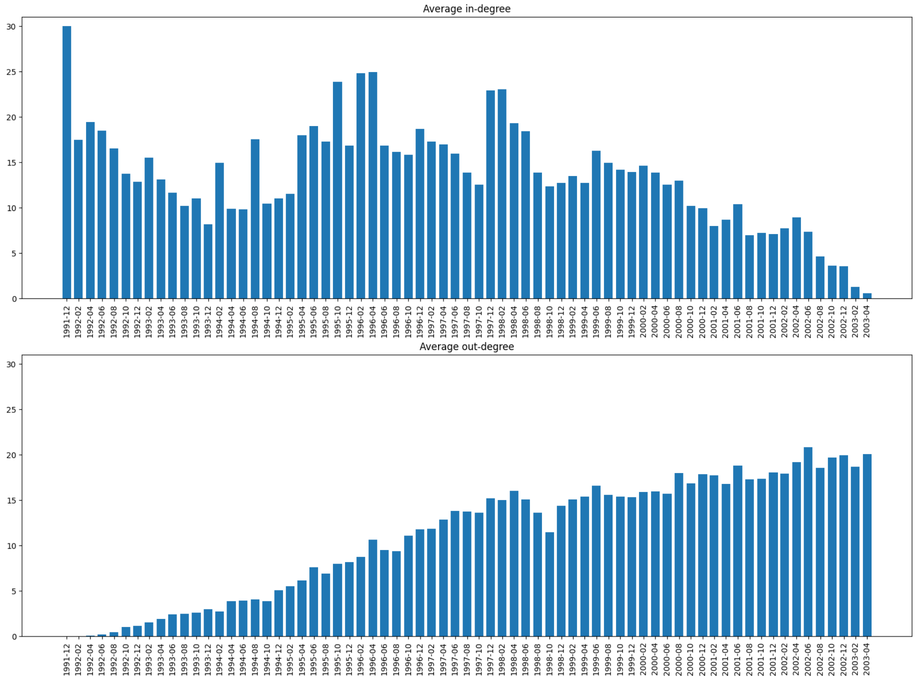

Figure 6: Average degrees in 2 months time from 1991 to 2003.

some embedding method in combination with a simple regression or classification model(depending on the task). We used similar approach in out study

As a baseline model we employed logistic regression classifier as a classical model. We trained it on three different node embeddings:

1. Text embeddings.
   We tried two different approaches of getting text embeddings:

   a) Text had been preprocessed using stemmer and stop words removal before applying bag of words vectorization.

   b) Text embeddings were obtained by averaging the embedding of each word in the abstract using glove-wiki-gigaword model with an embedding size of 50

2. Graph embeddings.
   We used the classic node2vec with default parameters, except that the dimension of the embedding space was 64 (instead of default 128)

The results of these methods can be found in the Table 2

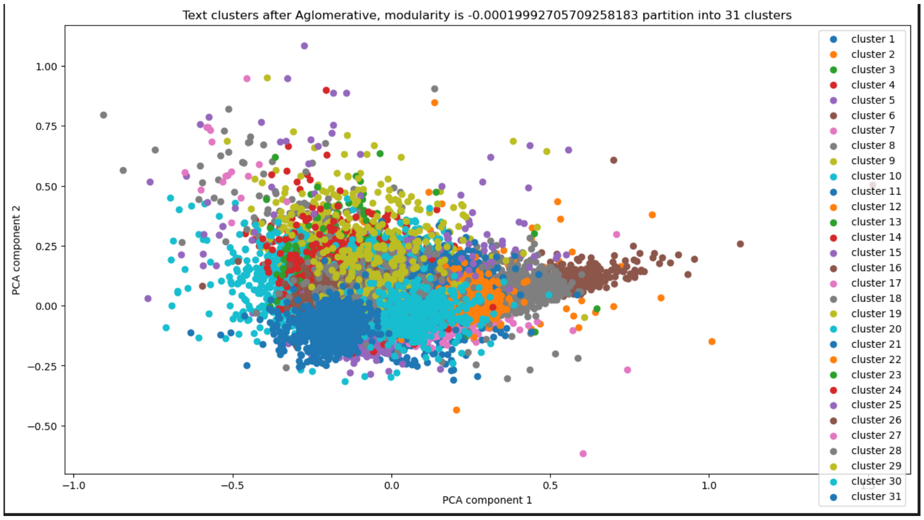

Figure 7: Visualization of clustering using Agglomerative Clustering algorithm.

Classification model	Embeddings	Metrics		
		Accuracy	F1 score	
			Macro	Weighted
Logistic Regression	Preprocessing + BoW	0.46	0.38	0.45
	glove-wiki-gigaword	0.44	0.16	0.27
	node2vec	0.53	0.33	0.43

Table 2: Baseline metrics.

4.3 Our solution

In search of a better solution, we tried GCN and GCNwLinear models with different types of node embeddings: random embeddings (learnable), text/graph-only embeddings and their concatenation (we tried using each of them separately, as well as in combination with each other). In addition, we searched through the following hyper parameters and settled on:

```
n_layers = 6
random_emb_size = 32
hidden_dim = 128
crop_df = False
tune_embs = True
add_reverse_edges = True
dropout=0.2
lr = 5e-4
node_emb_size = 64
```

Each parameter speaks for itself, except for `crop_df`, which is responsible for

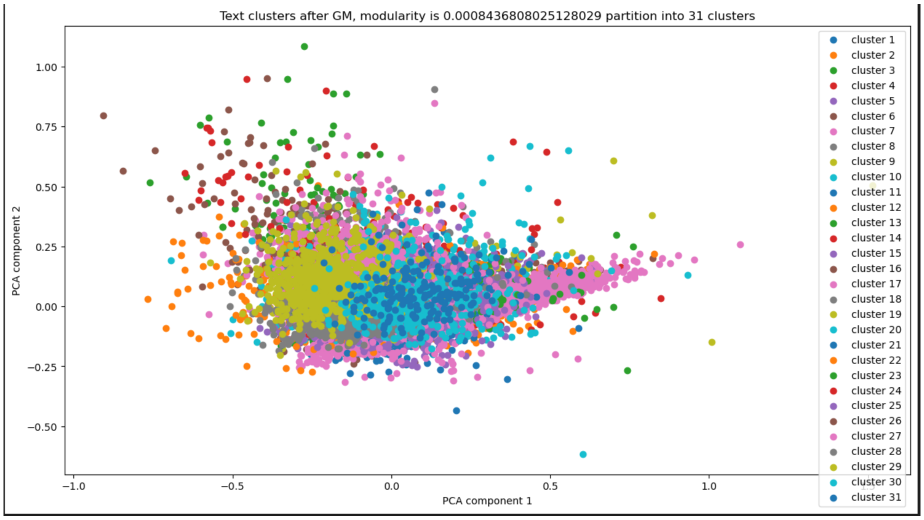

Figure 8: Visualization of clustering using Gaussian Mixture model.

removing the imbalance of classes in the data set by removing some samples. The idea was to feed the models an equal number of samples per class to minimize the risk of ignoring the class in the prediction, which was the case with the baseline models. The above is the best subset of these hyper parameters that we were able to find for the GCNwLinear model, which uses combination of text and node embeddings. The metrics achieved by this model are shown in Table 3.

Classification model	Embeddings	Metrics		
		Accuracy	F1 score	
			Macro	Weighted
GCNwLinear	Concat text and node	0.58	0.54	0.59

Table 3: Final metrics.

We achieved better overall metrics compared to our baseline, but one more prominent improvement is that this model does not ignore any classes in the sense that it decides to maximize the score by not predicting them at all. In Table 4 the difference of the baseline model and our model performance is shown with respect to this issue.

5 Conclusion

Unlike the other works, we implemented the combination of text embeddings and node embeddings to complete the task of predicting the citation rate. As we can see, on the chosen dataset, this method performed better than

Classification model	Class metrics			
	Class ID	Precision	Recall	F1 score
Baseline	0	0.54	0.90	0.68
	1	0.00	0.00	0.00
	2	0.25	0.04	0.07
	3	0.54	0.58	0.56
GCNwLinear	0	0.75	0.65	0.70
	1	0.28	0.36	0.32
	2	0.37	0.40	0.38
	3	0.75	0.75	0.75

Table 4: Comparison of baseline and our model on class metrics.

other methods from previous works on this subject. Not only were the metrics improved, but also the balance of predicted citation rates became better, not ignoring or overemphasizing any particular class. This goes to show that GNN models have a bright future regarding the task of predicting citation rates. As usual, those models should be tested more on different datasets in different areas of scientific publication to confirm their effectiveness and achieve better results.

References

[FDG20] Andrea Fronzetti Colladon, Ciriaco D'Angelo, and Peter Gloor. "Predicting the future success of scientific publications through social network and semantic analysis". In: *Scientometrics* 124 (July 2020), pp. 357–377. DOI: 10.1007/s11192-020-03479-5.

[Rom+21] Muhammad Roman et al. "Citation Intent Classification Using Word Embedding". In: *IEEE Access* PP (Jan. 2021), pp. 1–1. DOI: 10.1109/ACCESS.2021.3050547.

[Gao+24] Tianchen Gao et al. "Citation counts prediction of statistical publications based on multi-layer academic networks via neural network model". In: *Expert Systems with Applications* 238 (2024), p. 121634. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.121634. URL: https://www.sciencedirect.com/science/article/pii/S095741742302136X.

[Jr+24] Adilson Vital Jr. et al. *Predicting citation impact of research papers using GPT and other text embeddings*. 2024. arXiv: 2407.19942 [cs.DL]. URL: https://arxiv.org/abs/2407.19942.

[ZW24] Fang Zhang and Shengli Wu. "Predicting citation impact of academic papers across research areas using multiple models and early citations". In: *Scientometrics* 129 (June 2024), pp. 1–30. DOI: 10.1007/s11192-024-05086-0.