Predicting Citation Counts of Computer Science Papers from Microsoft's Academic Graph

Anton Abilov*
Connective Media
Cornell Tech
aa2776@cornell.edu

Eli Roussos*
Electrical and Computer Engineering
Cornell Tech
ekr43@cornell.edu

Jan Bernhard*
Computer Science
Cornell Tech
jhb353@cornell.edu

Abstract

The ever-increasing amount of scientific literature released to the public creates the need for a more efficient method to assess the importance of a piece of work at the time of its release. One popular answer has been to fit machine learning models to predict a paper's citation count as a placeholder metric for the paper's importance. We survey the various features used in those methods and apply them to bibliometric data from Microsoft's Academic Graph. To evaluate the features' predictive capability, we fit several different machine learning model types, such as neural networks and tree-based gradient boosting. The problem of predicting high or low citation counts is presented as a binary classification. The experiment results show that a paper's metadata yields better predictive capability compared to textual features retrieved from a paper's abstract, though neither method is robust in identifying papers with high citation reliably.

1 Introduction

Searching for academic papers can be an arduous process, requiring people to scour journal databases carefully for relevant content. This task is extraordinarily tedious because the number of papers published has steadily increased over the last couple of decades, as illustrated by Figure 1. The data presented in Figure 1 only considers peer-reviewed papers from journals and conferences in the field of computer science, which means it does not include the even larger number of papers published only on arXiv. Therefore, it is of interest to consider methods that help researchers isolate the subset of papers worth their time upon their release.

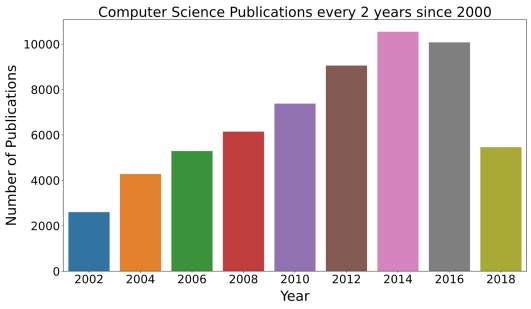


Figure 1: The number of CS papers published has increased significantly since the year 2000.

In this work we investigate whether it is possible to use machine learning methods to identify an important paper based on its abstract and metadata. The investigation is based on bibliometric data from Microsoft's Academic Graph [13, 19]. To confine the scope of our experiment, we only consider computer science-specific journal and conference papers since and including the year 2000. Similar to other experiments attempting to predict a paper's importance [20, 15, 1, 3], we assume that the citation count of a paper captures its

*The authors of this paper contributed to the presented work in equal parts. The Github link to this report's repository is `https://github.com/anton164/predicting-citation-counts`.

Applied Machine Learning Project Report. Fall 2020.

relevance and significance. Hence, it serves as the measure of importance in the context of this work. We chose this approach to make the results of this work more comparable to the similar experiments mentioned previously. This work investigates the following three hypotheses:

1. Based on a paper's metadata, it is possible to fit a model to predict whether a paper has a high or low citation count.
2. We can achieve a similar prediction accuracy with a model learned on text features retrieved from the paper's abstract.
3. Metadata and content features are complementary. Consequently, combining them improves the prediction accuracy.

This work's main contribution is a survey and evaluation of features used by the relevant literature to predict a paper's citation count.

2 Related Work

A common approach of evaluating a paper's popularity is to consider measures based on citation counts [2, 18]. Consequently, it is desirable to predict the citation count of a piece of work as early as possible in the research process.

A variety of traditional machine learning [20, 15, 14] as well as deep learning approaches [1, 9] aim to predict the citation count of a piece of literature. However, some of these methods tend to derive their predictive capabilities not from the paper's content but rather from the platform of publication and the author's prominence [6]. There is plenty of evidence suggesting that many more attributes of a paper correlate to the paper's citation count including, but not limited to, the paper's page count, number of authors, the content of the paper's title, and abstract [16]. Building on these insights, our approach aims to use that correlation to learn a model that provides predictions on a paper's citation count based on a combination of features that consider the paper's content and metadata. Similar to [20], we focus on papers published in the field of computer science.

Some approaches aim to phrase the citation count problem as a regression [1, 20, 14, 12] while others simplify the problem to that of a binary classification [3]. Given our objective to identify highly cited papers amongst most barely cited papers, we do not care about forecasting an exact citation count number. Consequently, we phrase our prediction problem as a binary classification.

[3] achieves a high binary classification performance using textual features derived from the papers' abstracts. This is surprising given that other methods found that correlation between a paper's citation count and content has been difficult to capture with a predictive model [6]. Consequently, we reimplement [3] approach as part of our investigation.

The binary classification methods [3, 15] tend to run their citation count prediction experiments on datasets on the scale of a few thousand papers, which is relatively small. In contrast, we deliberately experiment with a considerably larger dataset of around 60 thousand papers in this work.

Alternatives to the previously mentioned methods consider a given paper's citation count in the years after its publishing date to predict long-term performance [15], or a paper's visual appearance [9] to predict its future impact. To the best of our knowledge, [17] introduced the idea to consider a paper's appearance to assess its impact. However, [17] was presented and distributed as a joke paper in the context of a CVPR workshop. Our approach does not consider any paper specific citation count information as a feature, nor does it consider the paper's appearance or content beyond its abstract. The latter is not possible for us due to the nature of Microsoft's Academic Graph as the data source.

Lastly, the practice of using citation counts as a measure of research importance has to be taken with a grain of salt: evaluative bibliometrics used for ranking papers, journals and researchers in search engines such as Google Scholar have been recognized as an issue for the academic field, since it can incentivize chasing of metrics instead of pursuing genuinely innovative research [7].

3 Data Acquisition

The subject of our investigation is a dataset of five million academic papers extracted from the Microsoft Academic Graph [13, 19]. The dataset provides information common to a complete

BibTeX entry with additional information about a paper's importance, citation count, and fields of study. To reduce the dataset's size to a subset in order to accommodate the available hardware constraints, the first step of the data processing selects only the pieces of literature labeled as journal or conference papers with complete publisher and journal information. To further reduce the dataset size, we decided to focus on papers in the field of computer science published after the year 2000. The final dataset includes 61,019 academic paper entries.

4 Feature Extraction

The features created from the data's attributes are of two general types: textual features derived from a paper's abstract and features from a paper's metadata. We start this section by explaining the featurization of the citation count.

4.1 Yearly Citation Count Binning

To predict a paper's significance, we phrase the problem as a binary classification task: a paper is predicted to be either notable or insignificant. Consequently, it is necessary to bin the citation counts into two classes.

Firstly, it is necessary to normalize the citation count over time to make the feature comparable across papers published at different points in time. A paper's comparable citation count, $\bar{cc}_i$, is the paper's total citation count, cc_i, divided by the time number of years elapsed since paper's publication t_i, $\bar{cc}_i = \frac{cc_i}{t_i}$.

Secondly, it is necessary to bin $\bar{cc}_i$ into two different classes $y \in \{0, 1\}$. We use a Gaussian mixture model (GMM) to identify clusters in our data. The Elbow method via Kmeans informs our guess of the number of clusters to fit to the data. Next, we inspect the GMM's means and variances to see which clusters could form the target classes, y. By changing the clusters associated with each class, we can decide to be more or less strict with what we consider a highly cited paper.

4.2 Journal Rank and Publisher Rank

The first features considered to be predictive of a paper's citation count are "journal related factors", including which journal published the paper and who is the journal's publisher [16]. We first experimented with a one-hot encoding for this feature, but it led to a sparse feature matrix with poor predictive capabilities. Inspired by Yan et. al [20], we instead calculated the relative ranking of each journal and publisher based on their average citation counts $\bar{r}$ as follows:

$$\bar{r}_j = \frac{\sum_{k=0}^{M_j} cc_k}{M_j} \tag{1}$$

To compute $\bar{r}_j$, we gather all M_j papers for j journals or publishers. Subsequently, we assign a unique rank position based on $\bar{r}_j$. A higher rank means that on average the papers published by this journal or publisher received more citations. Journal and publisher rankings are calculated separately.

4.3 Author-specific Features

Additionally, we incorporate "author related factors", which have also been shown to hold predictive power [16]. However, in our dataset a paper might have up to 12 authors and performing one-hot encoding results in an extremely sparse feature matrix. To account for this we calculate two features: (1) "Author Prominence" - a binary feature which indicates whether one or more authors of the paper have been highly cited in other papers; (2) "Author Rank" - a rank feature that calculates the rank of all authors on the paper and averages them. The author prominence is dependent on the number of times the authors of the paper have been cited in other papers, c_{author}, and the average number of citations per paper within our dataset $barcc_{ave}$ as shown in Equation 2.

$$f(x) = \begin{cases} 0 & if \quad c_{author} \leq barcc_{ave}; \\ 1 & if\ 1 \leq c_{author} > barcc_{ave}; \end{cases} \tag{2}$$

Given the binary nature of the author prominence feature, it is unable to capture information about extremely successful authors. To that end, we added an *author rank* feature, which encodes additional

information about a paper's authors by averaging the authors' ranks. An author's rank is calculated using Equation 1, with the difference being that the citation count of the paper under investigation is excluded from the rank calculation. In addition to author prominence and author rank, we include *author count* as another possible feature, which yielded good results in [12]. It is calculated by counting the number of unique authors of the paper.

4.4 Page Count

According to [14], the page count of an academic paper is a significant feature used in predicting its citation count. The page count feature is calculated by subtracting the page number of the first page of a paper from the page number of the last page of a paper.

4.5 Publication Month

When attempting to predict a paper's citation counts using a neural network, Ruan et. al [12] found that a paper's publication month was one of the five most significant features. Consequently, we extract and one-hot-encode the publication month of each paper.

4.6 Textual Features

While some paper's claim [6] that textual features retrieved from a paper's content have little predictive capacity for citation counts, [3] found that the words within a paper's abstract yield useful features related to trending research topics.

To extract information from the paper's abstract, we tokenize and lemmatize the textual content using the *spacy* NLP library [8]. Tokens that match *spacy*'s notion of punctuation, pronouns, numbers, spacing or stopwords are removed. Then we create a vectorized representation of these tokens as a bag-of-words model. Instead of using a large sparse feature matrix encoding all frequent words in the abstracts, we found it to be beneficial to focus on the most *polarizing* words. Similar to [3], we start with considering only a fraction of the papers with the highest and lowest time normalized citation counts. The most polarizing words are those that occur significantly more often in the fraction of papers with high citation counts, h, compared to those with low citation counts, l, and vice-versa. We calculate the relative difference d as $d = 2\frac{h-l}{h+l}$. Based on this metric, we chose the words associated with the largest absolute differences. Our final selection consists of 50 word features.

5 Experiments

The objective of the experiment is to test whether metadata or textual content features are suited to predict a paper's association with the high or low citation count class.

5.1 Procedure

The experiment consists of three parts: in the first part of the experiment, a series of models are fit exclusively to the meta-data features; in the second part, the same set of models are fit to the textual content features only; in the third part, we combine the metadata and textual content features and repeat the fitting process.

5.2 Models

The selected models for this experiment are a linear support vector classifier (SVC), a multimodal Naive Bayes classifier, a random forest classifier, a multi-layer perceptron classifier, and lastly an XGBoost classifier. These models, bar XGBoost, were chosen as they are represented in the relevant literature [15, 3]. XGBoost was chosen as an additional classifier to provide a baseline result for a boosting-based ensemble model, as opposed to a bagging-based ensemble model such as a random forest.

We used the open-source machine learning libraries *scikit-learn* [11] and *XGBoost* [5] to obtain the results listed below.

Citation Count Threshold	Instances Per Class	
0.14997	15095	45924
2.06393	47059	13960
7.85759	**56198**	**4821**
25.2497	60103	916

Table 1: Yearly Citation Count Binning

Feature	Class 0			Class 1		
	min	**mean**	**max**	**min**	**mean**	**max**
Average Yearly Citation Count (**Target**)	0	1.12	7.86	7.86	22.9	1143.33
Journal Rank	1	659	1438	167	1137	1439
Publisher Rank	1	328	656	81	463	669
Author Rank	1	213	6227	463	744	6233
Number of Authors	1	2.9	11	1	3.2	9
PageCount	0	10.5	99	0	13	95
	Distribution					
AuthorProminence	35.8%			67.8%		

Table 2: Feature Distribution

5.3 Evaluation Metrics

For evaluating our predictive model we primarily use the F_1 score, which is a weighted average of the precision and recall values. Since our classes are imbalanced, the F_1 score is appropriate as it can account for the prior probabilities of our class labels [10]. The F_1 score of a model is given by: $F_1 = 2 \cdot \frac{P \cdot R}{P+R}$, where P is the precision and R is the recall of the model. In addition, we calculate the AUC score and accuracy of all test models for completeness. All metrics are calculated as defined on *scikit-learn* [2].

6 Results & Discussion

6.1 Descriptive Analysis

Through binning the yearly citation counts using a GMM, we arrive at four possible thresholds for separating the data into two classes (Table 1). As expected, there are far more papers with few citations in our dataset. We chose the third threshold of 7.85759 average annual citations since it is most suited to distinguishing high performing papers from low performing papers. Table 2 shows the distribution of our features associated with each value of the class to be predicted.

6.2 Significant Features

The correlation matrix between the extracted features and the target variable (Figure 2) show promising results. The matrix shows that the journal in which a paper is published has a strong correlation with the paper's citation counts. This implies that even though a paper may be written by a brilliant author, the author should seek to publish in a highly ranked journal in order to gain more citations.

In order to compare the relative importance of the selected features we use two methods. First, we rank the features using recursive feature elimination with cross-validation. Then, we use a "leave-one-out" approach inspired by [12]. We sequentially drop one feature and use the remaining features as a "leave-one-out-model". The performance of the "leave-one-out model" is compared with a model trained on all the features to infer whether a feature is significant.

[2] scikit-learn's model evaluation methods offer exact mathematical definitions: `https://scikit-learn.org/stable/modules/model_evaluation.html`

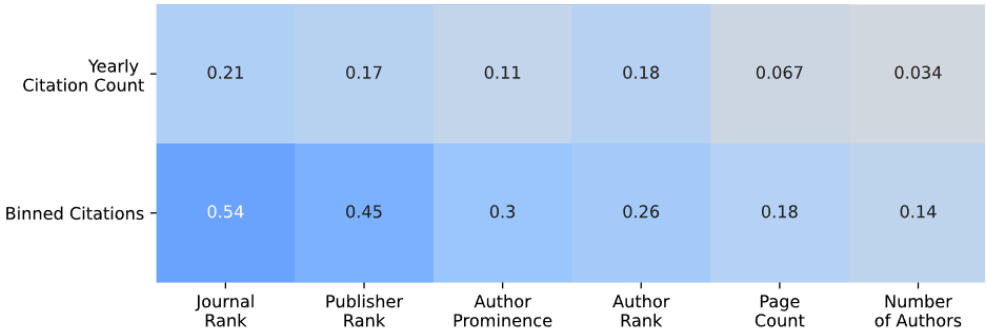
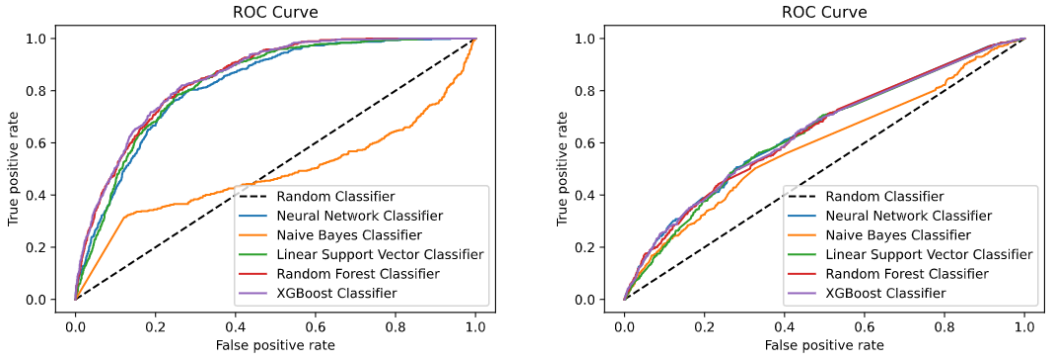


Figure 2: Binning significantly improved correlation between features and the prediction target.



(a) Models fitted to metadata features

(b) Models fitted to textual features

Figure 3: Model specificity vs sensitivity for varying classification threshold

The results of the feature selection show that Journal Rank, Publisher Rank, Number of Authors, Author Rank and Page Count should be included, while the other features can be removed. In the following experiments we use this new set of features.

6.3 Prediction Results

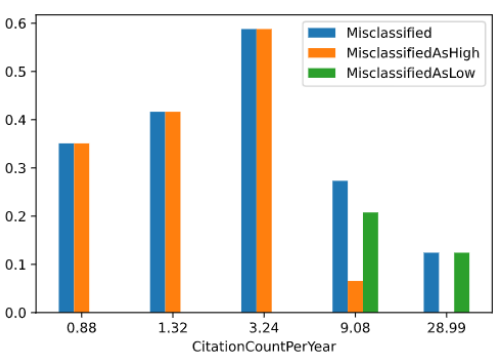
In this section, we report the results of the three parts of the experiment using the F1-score as key evaluation metric. To complement the F1-score we also report the models' ROC curves and AUC scores as additional metrics. To train our models we split the dataset into a training set with 80% of the papers, validation set of 10%, and a test set containing the remaining papers. The results of our models are calculated by taking the average scores on five random splits of the data.

To account for the imbalanced class distribution, we experimented with various sampling methods to create an equal ratio of data points for each class. Given that we have only a few thousand training samples for the highly cited papers class, we first tried oversampling using the Synthetic Minority Over-sampling Technique (SMOTE) [4]. This did not significantly improve the performance of our models, which can be explained by the feature distribution in Table 2. The classes do not have significantly distinguishing features from which it is possible to generate additional samples. For subsequent experiments we used random under-sampling.

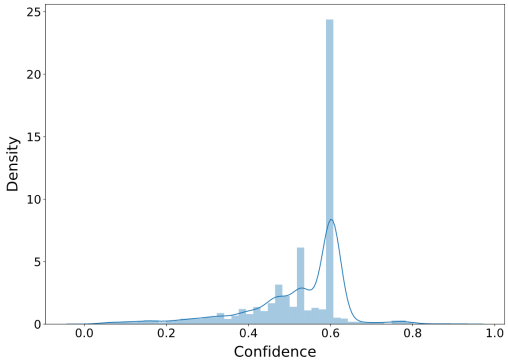
6.3.1 Predictions with Metadata Features

Even after hyperparameter-tuning, the classifiers are unable to achieve great performance on the test set. To better understand why this is the case, let us examine the model with the highest F1-score as indicated by the scores shown in Table 3; the XGBoost classifier.

The confusion matrix implies that the XGBoost classifier is unable to learn how to reliably identify highly cited papers. To investigate these misclassifications in more depth, we create five equally sized bins for average yearly citation count and plotted the misclassifications accordingly. Figure 4a shows



(a) Misclassifications distributed by citation counts



(b) *Low* citation count-class prediction confidence distribution for XGBoost model fitted to textual features.

Figure 4: Error analysis

Classifier	F1-score	Accuracy	ROC AUC
Uniform baseline	**50.0%**	50.0%	50.0%
Most frequent baseline	0.0%	**91.3%**	50.0%
Neural Network	31.1%	66.5%	82.8%
Naive Bayes	25.2%	81.7%	49.8%
Linear SVM	32.5%	68.7%	84.0%
Random Forest	33.2%	70.5%	**85.6%**
XGBoost	**33.6%**	72.0%	85.4%

(a) Fitted to metadata features

F1-score	Accuracy	ROC AUC
50.0%	50.0%	50.0%
0.0%	**91.3%**	50.0%
19.3%	62.7%	63.6%
18.7%	59.8%	61.9%
20.5%	66.2%	64.5%
19.1%	61.2%	64.9%
21.6%	69.6%	**65.3%**

(b) Fitted to textual features

Table 3: Prediction performance of various models.

that our error rate is higher when closer to the threshold, but around 35% of misclassifications occur close to 0 citations. Table 5 shows five samples of papers with 0 citations that were misclassified as high. The pattern seems clear: predicting average citation counts purely based on these features incorrectly implies that a paper is likely to receive many citations when it's published in a renowned journal. This is further exacerbated if it is written by prominent authors.

Reflecting on the performance of our classifiers, we infer that an accurate citation count prediction using our constructed features is not feasible. The resulting models have high bias and thus underfit the data. Due to the high false negative rate (19% with XGBoost) our trained model is not suitable for assisting researchers in filtering out noise when considering newly published papers.

6.3.2 Predictions with Textual Features

Using the "polarizing words" method described earlier, we extracted the most polarizing words from 10% of the highest and lowest papers (by citation counts).

		Predicted class		
		Class 0	Class 1	Total
True Class	Class 0	4092	1535	5627
	Class 1	89	386	475
	Total	4181	1921	N

Table 4: Confusion Matrix for the XGBoost Classifier

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
Ave. Yearly Citation Count	0.0	0.0	0.0	0.0	0.0
JournalRank	995	1114	1126	1384	866
PublisherRank	369	369	483	478	478
# of Authors	5	3	2	5	3
AuthorRank	1138	422	2	819	324
PageCount	8	14	4	2	0
Predicted Class	1	1	1	1	1
Confidence	73%	72%	56%	79%	60%

Table 5: Example of five misclassified papers.

Most polarizing words at a 10% threshold

Top 10 (positive correlation with the target): relay, empirical, brain, availability, http, classifier, protein, survey, literature, we

Bottom 10 (negative correlation with the target): library, book, calculation, page, table, but, so, matlab, firstly, according

As indicated in Figure 4b and Table 3b, the results from training a model purely on textual features are far below the baseline. This suggests that this method is not sufficient for classifying highly cited papers. This directly contradicts the results obtained in [3]. Nevertheless, we found it noteworthy that papers which mention MATLAB in their abstract correlate with a lower citation count.

6.3.3 Predictions with Metadata & Textual Features

In the final experiment, we combined the metadata and textual features. To find an appropriate scaling of the different features, we experimented with MinMax and Standard Scaling. The best model trained on these features, XGBoost, obtained an F1-score of 31.4% which was slightly lower than the optimal model trained on metadata features. This confirms that the textual features we extracted have little predictive capability.

7 Limitations & Future Work

There are a few limitations in this study which if addressed could lead to better results. First of all, the dataset does not contain relevant information about each paper at the time of publication - it is merely a recent snapshot. Hence, the metadata features of a paper could incorrectly indicate that its authors are ranked highly, even though when the paper was initially published the authors had a low ranking. This issue extends to the journal rank and publisher rank features as well.

Furthermore, our experiments were constrained by the scope of the metadata features derived solely based on the snapshot. Surely, there is more information available about each entity which could have predictive capabilities, such as an author's h-index, an author's job-title at the time of publication or a journal's impact factor. We believe that extending our dataset to incorporate this additional information, some of which can be obtained through the Microsoft Academic Graph at an additional cost, could be viable next steps in predicting a paper's citation counts more reliably. It may be worth investigating whether ensemble of models that classify based on varying feature subsets take better advantage of a dataset with extended features.

8 Conclusion

In our experiment, we surveyed various features derived from a paper's metadata and abstract. Across the board, the models trained on the metadata features outperformed those trained on the textual abstract features. Overall, we found no evidence that textual features from a paper's abstract provided any additional signal. It will take further feature engineering to arrive at a set of classification models that can be used to presort papers into relevant and irrelevant categories at the time of their release.

References

[1] A. Abrishami and S. Aliakbary. Predicting citation counts based on deep neural network learning techniques. *Journal of Informetrics*, 13(2):485–499, sep 2019. ISSN 18755879. doi: 10.1016/j.joi.2019.02.011. URL http://arxiv.org/abs/1809.04365http://dx.doi.org/10.1016/j.joi.2019.02.011.

[2] D. W. Aksnes, L. Langfeldt, and P. Wouters. Citations, citation indicators, and research quality: An overview of basic concepts and theories. *SAGE Open*, 9(1):2158244019829575, 2019. doi: 10.1177/2158244019829575. URL https://doi.org/10.1177/2158244019829575.

[3] T. Baba and K. Baba. Citation Count Prediction Using Non-technical Terms in Abstracts. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10960 LNCS, pages 366–375. Springer International Publishing, 2018. ISBN 9783319951614. doi: 10.1007/978-3-319-95162-1_25. URL http://dx.doi.org/10.1007/978-3-319-95162-1{_}25http://link.springer.com/10.1007/978-3-319-95162-1{_}25.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 2002. ISSN 10769757. doi: 10.1613/jair.953.

[5] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939785. URL http://doi.acm.org/10.1145/2939672.2939785.

[6] R. Daniel. Predicting citation counts, Jun 2014. URL https://www.researchtrends.com/issue-37-june-2014/predicting-citation-counts/.

[7] J. Goldenfein, S. Benthall, D. S. Griffin, and E. Toch. Private Companies and Scholarly Infrastructure - Google Scholar and Academic Autonomy. *SSRN Electronic Journal*, pages 1–10, 2019. ISSN 1556-5068. doi: 10.2139/ssrn.3476911.

[8] M. Honnibal and I. Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. 2017.

[9] J.-B. Huang. Deep Paper Gestalt. *arXiv*, pages 1–7, dec 2018. URL http://arxiv.org/abs/1812.08775.

[10] L. A. Jeni, J. F. Cohn, and F. De La Torre. Facing imbalanced data - Recommendations for the use of performance metrics. In *Proceedings - 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, ACII 2013*, 2013. ISBN 9780769550480. doi: 10.1109/ACII.2013.47.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[12] X. Ruan, Y. Zhu, J. Li, and Y. Cheng. Predicting the citation counts of individual papers via a BP neural network. *Journal of Informetrics*, 2020. ISSN 18755879. doi: 10.1016/j.joi.2020.101039.

[13] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-J. P. Hsu, and K. Wang. An Overview of Microsoft Academic Service (MAS) and Applications. In *Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion*, pages 243–246, New York, New York, USA, 2015. ACM Press. ISBN 9781450334730. doi: 10.1145/2740908.2742839. URL http://dl.acm.org/citation.cfm?doid=2740908.2742839.

[14] B. Sohrabi and H. Iraj. The effect of keyword repetition in abstract and keyword frequency per journal in predicting citation counts. *Scientometrics*, 110(1):243–251, 2017. ISSN 15882861. doi: 10.1007/s11192-016-2161-5.

[15] Z. Su. Prediction of future citation count with machine learning and neural network. In *2020 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, pages 101–104. IEEE, apr 2020. ISBN 978-1-7281-6067-2. doi: 10.1109/IPEC49694.2020.9114959. URL https://ieeexplore.ieee.org/document/9114959/.

[16] I. Tahamtan, A. Safipour Afshar, and K. Ahamdzadeh. Factors affecting number of citations: a comprehensive review of the literature. *Scientometrics*, 107(3):1195–1225, 2016. ISSN 15882861. doi: 10.1007/s11192-016-1889-2.

[17] C. Von Bearnensquash. Paper Gestalt. *CVPR workshop*, pages 345–349, 2010.

[18] L. Waltman. A review of the literature on citation impact indicators. *Paper presented at the Annual Meeting of the Mid-S*, pages 1–71, jul 2015. ISSN 0968-4891. URL `http://eric.ed.gov/?id=ED448174http://arxiv.org/abs/1507.02099`.

[19] K. Wang, Z. Shen, C. Huang, C.-H. Wu, D. Eide, Y. Dong, J. Qian, A. Kanakia, A. Chen, and R. Rogahn. A Review of Microsoft Academic Services for Science of Science Studies. *Frontiers in Big Data*, 2(December):1–16, dec 2019. ISSN 2624-909X. doi: 10.3389/fdata.2019.00045. URL `https://www.frontiersin.org/article/10.3389/fdata.2019.00045/full`.

[20] R. Yan, J. Tang, X. Liu, D. Shan, and X. Li. Citation count prediction: Learning to estimate future citations for literature. In *International Conference on Information and Knowledge Management, Proceedings*, pages 1247–1252. Association for Computing Machinery, 2011. ISBN 9781450307178. doi: 10.1145/2063576.2063757.