Simplifying Impact Prediction for Scientific Articles

Thanasis Vergoulis
IMSI, ATHENA RC
vergoulis@athenarc.gr

Ilias Kanellos
IMSI, ATHENA RC
ilias.kanellos@athenarc.gr

Giorgos Giannopoulos
IMSI, ATHENA RC
giann@athenarc.gr

Theodore Dalamagas
IMSI, ATHENA RC
dalamag@athenarc.gr

ABSTRACT

Estimating the expected impact of an article is valuable for various applications (e.g., article/cooperator recommendation). Most existing approaches attempt to predict the exact number of citations each article will receive in the near future, however this is a difficult regression analysis problem. Moreover, most approaches rely on the existence of rich metadata for each article, a requirement that cannot be adequately fulfilled for a large number of them. In this work, we take advantage of the fact that solving a simpler machine learning problem, that of classifying articles based on their expected impact, is adequate for many real world applications and we propose a simplified model that can be trained using minimal article metadata. Finally, we examine various configurations of this model and evaluate their effectiveness in solving the aforementioned classification problem.

KEYWORDS

scientific impact, machine learning, classification

1 INTRODUCTION

Predicting the attention a scientific article will attract in the next few years by other articles, i.e., estimating its expected *impact*[1], is very useful for many applications. For example, consider a recommendation system, which suggests articles to researchers based on their interests. Due to the large growth rate in the number of published research works [9], a large number of articles will be retrieved for almost any subject of interest. However, not all of them will be of equal importance. The recommendation system could leverage the expected impact of papers to suggest only the most important works to the user and avoid overwhelming her with a large number of trivial options. The benefits would be similar for other relevant applications, such as expert finding, collaboration recommendation, etc.

Several approaches, which attempt to predict the exact number of citations articles will receive in the next few years, have been proposed in the literature (see Section 4 for indicative examples). However, this is an extremely difficult regression analysis problem, due to the many factors (some of which are hard to quantify) that may affect the impact of an article (details in Section 2.2). Fortunately, in practice, for many applications, knowing the exact number of future citations is not critical. For instance, in the case of the recommendation system, it is important that the system distinguish the 'impactful' works from those that are of lesser importance; all impactful works will be interesting regardless of the exact number of citations they will receive.

In addition, most existing approaches rely on rich article metadata (e.g., authors, venue, topics). Unfortunately, the available information for many articles in the relevant data sources (e.g., Crossref) is erroneous or incomplete, complicating the learning process of such approaches and creating risks for their effectiveness. Moreover, even when the required metadata are available, the generation of the corresponding machine learning features from them may be extremely time-consuming or even difficult to be implemented (details in Section 2.3).

In this work, our objective is to take advantage of the previous observations in an attempt to guide and facilitate the work of researchers and developers working on applications that can benefit from predicting the expected impact of scientific articles. In particular, we propose a simplified machine learning approach which is based on the binary classification of articles in two categories ('impactful' / 'impactless') according to their expected impact. In addition, we propose the use of a particular set of features that rely on minimal metadata for each article (only its publication year and its previous citations). We argue that this simpler approach is adequate, significantly easier to implement, and can benefit many applications that require the estimation of the expected impact of articles. Finally, we perform experiments to investigate the effectiveness of this approach using various well-established classifiers. In our experimental setup we seriously take into consideration the fact that our problem is imbalanced by nature, both to carefully select the appropriate evaluation measures and to examine some classification approaches that are particularly tailored to such scenarios.

2 OUR APPROACH

2.1 Preliminaries

Scientific articles always include a list of references to other works and the referenced articles describe work related to the referencing article (e.g., preliminaries, competitive approaches). As a result, the inclusion of an article in the reference list of another (i.e., the one *citing* it) implies that the latter gives credit to the former[2]. Based on this view, counting the number of distinct articles that include an article of interest in their reference list (i.e., counting its *citations*) is considered to be an indicator of its impact in the scientific community. Of course, there are also many other aspects of scientific impact [3], however the focus of this work is on this type of citation-based expected impact. In particular, we focus on the *expected impact* of an article at a given time point, which can be defined as follows:

[1] Since scientific impact has several aspects [3], the term can be defined in diverse ways. In this work, we focus on the definition provided in Section 2.1.

[2] Note that the "amount" of credit may be significantly different for each referenced work and that, in some cases, it may also have a negative sign (when the referencing work criticizes the referenced one).

Definition 2.1 (Expected Article Impact). Consider an article a and a time point t. Then, $i(a, t)$, the (expected) impact of a at t, is calculated as the number of citations that a will receive during the period $[t, t + y]$, where y is a problem parameter, which defines a *future period* of interest.

It should be noted that the problem parameter y can be configured based on the characteristics of the dataset used. The optimal option typically depends on the citation dynamics of the scientific fields covered by the dataset. However, $y = 3$ or $y = 5$ are two reasonable and very common configurations. Finally, it should be highlighted that the expected impact of an article can only be measured in retrospect, i.e., by monitoring the citations that the article receives y years after the time point of reference.

2.2 Problem definition

Considering the expected impact of articles can be useful for many applications. This is why there is a line of work of methods that attempt to predict the exact impact of each article, i.e., the exact number of citations it is going to receive in the following few years (see Section 4). However, this is a difficult regression analysis problem for many reasons. First of all, there are many factors that may affect the number of citations an article will receive in the future. These factors are related to the quality of the work, the hype of its topic, the prestige of its authors or its venue, the dissemination effort that will be made in social media, to name only a few. Also, to make matters worse, many of these factors cannot be easily quantified without losing important information (e.g., due to dimensionality reduction reasons in one-hot encodings), affecting the accuracy of the approaches.

Additionally, in practice, many of the aforementioned applications do not require the prediction of the exact number of future citations for each article. It is sufficient for them to simply distinguish between 'impactful' (to-be) and 'impactless' articles. This type of problem is easier and, thus, a traditional classification approach is likely to achieve adequate effectiveness in solving it. Hence, in this work, we focus on a binary impact-based article classification problem that can be formulated as follows:

Definition 2.2 (Impact-based article classification). Consider a collection of scientific articles A and a time point t and let $\bar{i} = \sum_{a \in A} i(a, t)/|A|$. Then, the objective is to classify each $a \in A$ in one of two classes: in the class of 'impactful' articles, if $i(a, t) > \bar{i}$ and to the class of 'impactless' articles, otherwise.

In other words, our objective is to identify articles that receive an above-average number of citations, to classify them as 'impactful' and the rest as 'impactless'. Note that this intuitive distinction is equivalent with the first iteration of the Head/Tail Breaks clustering algorithm, which is tailored for heavy tailed distributions, like the citation distribution of articles [2] (a small number of articles receive an extremely large number of citations).

An important matter that should be highlighted is that this classification problem is *imbalanced* by nature. Due to the fact that the citation distribution of articles is long-tailed, most articles have an impact (i.e., number of citations) well below average. Consequently, the class of 'impactful' articles will always be a minority in the collection (the so-called 'head' of the citation distribution). This is important for two reasons; first of all, it affects the correct choice of evaluation measures in the experimental setup. For example, using the accuracy (i.e., the ratio of true positives to the complete set) is problematic: a trivial classifier that would always assign all articles to the 'impactless' class will always achieve a good performance according to this measure. For this reason, alternative measures like precision, recall, and F1 of the minority class (i.e., the class of 'impactful' articles) should be used instead. Unfortunately, part of the previous literature (e.g., [18]) overlooks this issue making it difficult to evaluate the real effectiveness of the corresponding proposed approaches.

2.3 The proposed feature selection

Many existing machine learning approaches rely on the existence of various article metadata such as its publication year, author list, venue, main topics, citations etc. Although nowadays a large portion of such data becomes available through open scholarly graphs [6, 15] or datasets (e.g., DBLP, Crossref), there are many articles for which important information is erroneous, incomplete, or even completely missing. The main reason for this is that many such datasets are created by automatically harvesting, cleaning, and integrating data from heterogeneous (and sometimes noisy) primary sources.

However, even when all the required metadata are available, in many cases the generation of the desired machine learning features involves time-consuming aggregations and other processing tasks and may also be difficult to implement. For example, a number of data cleaning issues arise, for approaches using author-based features since author names have to be disambiguated in the case of synonyms, or different spellings across publication venues. Similarly, venue names might be recorded with different forms (e.g., acronyms vs. full names). Such issues affect the overall quality and, hence, the utility of these metadata.

It is evident that, relying on rich article metadata is an important limitation for any machine learning approach to predict the expected impact of articles. On the other hand, an article's publication year is a basic information that is available in the vast majority of cases. As an indicative example, in the Crossref public data file of March 2020[3], only 7.85% of the records were missing this information. Moreover, due to the Initiative for Open Citations[4] (I4OC), an increasing number of publishers (with Elsevier being the most recent one) are committed to openly provide the reference lists of their articles. As a result, the majority of citation data are now available in open scholarly datasets (e.g., in Crossref). To summarize, the citations and the publication years of scientific articles are readily available data.

Based on the above, we propose a set of features that can be easily calculated using article citations and publication years. In particular, we calculate the following:

- *cc_total*: The total number of citations ever received by the article (i.e., its 'citation count').
- *cc_1y*: Citations received by the article in the last year.
- *cc_3y*: Citations received by the article in the last 3 years.
- *cc_5y*: Citations received by the article in the last 5 years.

The intuition behind these features is based on the idea of preferential attachment [2] and of its time-restricted version used in recent impact-based article ranking approaches [8]: articles that are likely to be highly cited in the following few years are most likely those, which were intensively cited in the recent past.

It should be noted that, although the minimum value of the features is zero in all cases, the largest value of each of them could be very diverse. This is why it is a good practice to normalize them before using them as input to the classifier.

[3] https://doi.org/10.13003/83B2GP
[4] https://i4oc.org/

Sample set	Samples	Impactful samples
PMC 2011 – 2013 (3 years)	229, 207	57, 016 (24.88%)
PMC 2011 – 2015 (5 years)	229, 207	61, 898 (27.01%)
DBLP 2011 – 2013 (3 years)	1, 695, 533	387, 506 (22.85%)
DBLP 2011 – 2015 (5 years)	1, 695, 533	339, 351 (20.01%)

Table 1: Used sample sets

Classifier	Examined parameter values
LR & cLR	'max_iter': 60, 80, 100, 120, 140, 160, 180, 200, 220, 240 'solver': 'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'
DT & cDT	'max_depth': 1 – 32 'min_samples_split': 2, 5, 10, 20, 50, 100, 200 'min_samples_leaf': 1, 4, 7, 10
RF & cRF	'max_depth': 1, 5, 10, 50 'n_estimators': 100, 150, 200, 250, 300 'criterion': 'gini', 'entropy' 'max_features': 'log2', 'sqrt'

Table 2: Parameter values examined per classifier.

3 EVALUATION

3.1 Setup

Datasets. For our experiments, we collected citations and publication years for scientific articles from two sources:

- *PMC*: The data were gathered from NCBI's PMC FTP directory[5] and are relevant to 1.12 million open access scientific articles from life sciences published between 1896 and 2016. Moreover, we removed data from the last year (they were incomplete, not the entire year was represented).
- *DBLP*: The data were collected from AMiner's DBLP Citation Network dataset[6] [19] and are relevant to 3 million articles published between 1936 and 2018. Moreover, we removed data from the last two, incomplete years.

To create the labeled samples required for our analysis, we follow the hold-out evaluation approach [7]: For each dataset we select the year $t = 2010$ as a (virtual) present year and we split the dataset in two parts: the first one (articles published until 2010, with 2010 included) to calculate the feature vectors described in Section 2.3 for all included articles; the second one to calculate the label for each sample, based on its future citations (see Section 2.2). We set $y = 3$ and $y = 5$ for the article impact future period (see Section 2.1), which corresponds in both our datasets to the periods $2011 - 2013$, and $2011 - 2015$, respectively. Table 1 summarizes the statistics of the sample sets that have been created based on the aforementioned process.

Classifiers. We selected to use a set of well-known classifiers, along with their cost-sensitive versions[7]. The reason we selected to include cost-sensitive versions is because they target the problem of imbalanced learning by using different misclassification costs for samples of different classes [5]. As a result, we have configured and evaluated the following classification methods:

- *LR*: Logistic regression
- *cLR*: Cost-sensitive logistic regression
- *DT*: Decision trees
- *cDT*: Cost-sensitive decision trees
- *RF*: Random forest

[5] ftp://ftp.ncbi.nlm.nih.gov/pub/pmc
[6] https://aminer.org/citation
[7] We used Scikit-learn's 'balanced' mode for $class\_weight$ to automatically adjust weights inversely proportional to class frequencies in the input data.

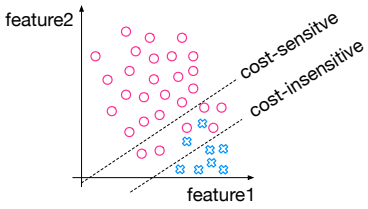

Figure 1: Toy example showcasing why cost-sensitive approaches may achieve worse precision.

- *cRF*: Cost-sensitive random forest

For all methods we used their Scikit-learn [16] implementations and we have followed a two-fold, exhaustive grid search approach to identify the optimal values of their parameters according to the precision, recall, and F1 of the minority class. Table 2 summarizes the parameter space examined, while Tables 5 & 6 in the Appendix enlist all the identified optimal configurations. Each optimal configuration is named as $[classifier]_{[measure]}$, where $[classifier]$ is the name of the corresponding classifier (e.g., LR, cLR) and $[measure]$ stands for the evaluation measure for which the configuration is optimal (e.g., 'prec' for precision).

3.2 Results

Because of the imbalanced nature of the classification problem we study, it is very important to carefully select the measures that will be used for the evaluation of the effectiveness of the examined approaches. For example, as it was discussed in Section 2.2, *accuracy* that is commonly used for generic classification approaches, is not a good option, since it is mostly affected by the misclassification of samples from the majority class. However, in most imbalanced problems, like the one we have here, the minority class has the most importance.

Therefore, we do not report the accuracy of the examined approaches. In any case, all configurations achieved accuracy between 0.73 and 0.99. Following the best practices for the evaluation of imbalanced classification approaches, we instead measure the precision, recall, and F1 of the minority class. We indicatively report the same measures for the majority class, as well. However our main objective is to perform well according to the measures calculated for the minority class. Note that, each of these three measures may be preferable for different applications.

Tables 3b & 4b summarize the results of the performed experiments. The results are very similar for both data sets (PMC and DBLP) and for both values of the parameter y. A general observation is that, when we focus on precision, cost-insensitive classification approaches perform adequately well and, thus, there is no need to work with cost-sensitive versions. However, the same experiments highlight that the latter approaches can significantly improve the effectiveness based on the recall and F1.

This behavior is not surprising: By default, in several classifiers, the optimization process targets at accuracy maximization, since all samples equally contribute to the loss function to be minimized. Consequently, in areas of the hyperspace where the samples of different classes are not easily separable, the samples of the majority class are favored (i.e., correctly classified) due to their dominance in numbers. Consider, for instance, the two minority class samples (cross marks) and the six majority class ones (cyclic marks) between the two alternative hyperplanes of the toy example in Figure 1: Classifying all of them to the majority class would induce 3 times less cost to the classifier than classifying

Classifier	Precision (impactful|rest)	Recall (impactful|rest)	F1 (impactful|rest)
LR_{prec}	**0.85**|0.79	0.23|0.99	0.36|0.88
LR_{rec}	**0.85**|0.79	0.23|0.99	0.36|0.88
LR_{f1}	**0.85**|0.79	0.23|0.99	0.36|0.88
cLR_{prec}	0.57|0.85	0.52|0.87	0.55|0.86
cLR_{rec}	0.57|0.85	0.52|0.87	0.55|0.86
cLR_{f1}	0.57|0.85	0.52|0.87	0.55|0.86
DT_{prec}	0.66|0.82	0.38|0.93	0.48|0.87
DT_{rec}	0.66|0.82	0.38|0.93	0.48|0.87
DT_{f1}	0.66|0.82	0.38|0.93	0.48|0.87
cDT_{prec}	0.60|0.85	0.52|0.89	**0.56**|0.87
cDT_{rec}	0.50|0.87	0.63|0.79	**0.56**|0.83
cDT_{f1}	0.52|0.86	0.60|0.81	0.55|0.84
RF_{prec}	0.70|0.82	0.38|0.95	0.50|0.88
RF_{rec}	0.71|0.82	0.37|0.95	0.48|0.88
RF_{f1}	0.71|0.82	0.36|0.95	0.48|0.88
cRF_{prec}	0.56|0.85	0.53|0.86	0.54|0.85
cRF_{rec}	0.47|0.87	**0.65**|0.76	0.55|0.81
cRF_{f1}	0.48|0.87	**0.65**|0.77	0.55|0.81

(a) PMC

Classifier	Precision (impactful|rest)	Recall (impactful|rest)	F1 (impactful|rest)
LR_{prec}	**0.97**|0.82	0.25|1.00	0.39|0.90
LR_{rec}	0.96|0.82	0.26|1.00	0.40|0.90
LR_{f1}	0.96|0.82	0.25|1.00	0.40|0.90
cLR_{prec}	0.70|0.88	0.57|0.93	0.63|0.90
cLR_{rec}	0.70|0.88	0.57|0.93	0.63|0.90
cLR_{f1}	0.71|0.88	0.56|0.93	0.63|0.90
DT_{prec}	0.80|0.88	0.55|0.96	**0.65**|0.92
DT_{rec}	0.72|0.89	0.61|0.93	0.61|0.91
DT_{f1}	0.72|0.89	0.61|0.93	0.61|0.91
cDT_{prec}	0.58|0.92	0.74|0.84	**0.65**|0.88
cDT_{rec}	0.52|0.93	**0.79**|0.78	0.63|0.85
cDT_{f1}	0.58|0.92	0.75|0.84	**0.65**|0.88
RF_{prec}	0.72|0.88	0.56|0.94	0.63|0.91
RF_{rec}	0.72|0.88	0.56|0.94	0.63|0.91
RF_{f1}	0.77|0.87	0.54|0.95	0.63|0.91
cRF_{prec}	0.64|0.89	0.63|0.89	0.64|0.89
cRF_{rec}	0.57|0.92	0.76|0.83	**0.65**|0.87
cRF_{f1}	0.58|0.92	0.76|0.84	0.65|0.88

(b) DBLP

Table 3: Precision, recall, and F1 based on future citations in [2011-2013] (3 years). Configurations in Tables 5 & 6.

Classifier	Precision (impactful|rest)	Recall (impactful|rest)	F1 (impactful|rest)
LR_{prec}	**0.89**|0.78	0.26|0.99	0.40|0.87
LR_{rec}	**0.89**|0.78	0.26|0.99	0.40|0.87
LR_{f1}	**0.89**|0.78	0.25|0.99	0.39|0.87
cLR_{prec}	0.60|0.82	0.49|0.88	0.54|0.85
cLR_{rec}	0.60|0.82	0.48|0.88	0.54|0.85
cLR_{f1}	0.60|0.82	0.49|0.88	0.54|0.85
DT_{prec}	0.75|0.81	0.38|0.95	0.50|0.87
DT_{rec}	0.75|0.80	0.35|0.96	0.48|0.87
DT_{f1}	0.75|0.81	0.39|0.95	0.51|0.87
cDT_{prec}	0.60|0.82	0.49|0.88	0.54|0.85
cDT_{rec}	0.50|0.84	0.61|0.78	0.55|0.81
cDT_{f1}	0.53|0.84	0.60|0.81	**0.56**|0.82
RF_{prec}	0.72|0.80	0.37|0.95	0.49|0.87
RF_{rec}	0.73|0.81	0.41|0.95	0.53|0.87
RF_{f1}	0.74|0.81	0.41|0.95	0.52|0.87
cRF_{prec}	0.57|0.82	0.49|0.86	0.52|0.84
cRF_{rec}	0.50|0.84	**0.61**|0.77	0.55|0.81
cRF_{f1}	0.50|0.84	**0.61**|0.77	0.55|0.81

(a) PMC

Classifier	Precision (impactful|rest)	Recall (impactful|rest)	F1 (impactful|rest)
LR_{prec}	0.96|0.84	0.24|1.00	0.39|0.91
LR_{rec}	0.96|0.84	0.24|1.00	0.39|0.91
LR_{f1}	**0.97**|0.84	0.24|1.00	0.38|0.91
cLR_{prec}	0.70|0.90	0.61|0.93	**0.65**|0.92
cLR_{rec}	0.73|0.90	0.58|0.94	**0.65**|0.92
cLR_{f1}	0.70|0.90	0.60|0.93	**0.65**|0.92
DT_{prec}	0.87|0.87	0.42|0.98	0.56|0.92
DT_{rec}	0.73|0.90	0.56|0.95	0.63|0.92
DT_{f1}	0.77|0.89	0.52|0.96	0.62|0.92
cDT_{prec}	0.59|0.93	0.72|0.88	**0.65**|0.90
cDT_{rec}	0.47|0.94	**0.82**|0.77	0.60|0.85
cDT_{f1}	0.59|0.93	0.72|0.88	**0.65**|0.90
RF_{prec}	0.83|0.89	0.52|0.97	0.64|0.93
RF_{rec}	0.74|0.90	0.56|0.95	0.64|0.92
RF_{f1}	0.80|0.90	0.56|0.96	0.66|0.93
cRF_{prec}	0.62|0.91	0.66|0.90	0.64|0.91
cRF_{rec}	0.59|0.91	0.67|0.89	0.63|0.90
cRF_{f1}	0.55|0.93	0.76|0.84	0.64|0.89

(b) DBLP

Table 4: Precision, recall, and F1 based on future citations in [2011-2015] (5 years). Configurations in Tables 5 & 6.

them to the minority class. In this way the cost-insensitive classifier also achieves good precision for the minority class (no false positives in this example). The drawback is that this results in many false negatives for the minority class (the most important one). Cost-sensitive approaches alleviate this issue improving the recall and F1 of the minority class, with the counter-effect of a larger number of false positives for the minority class.

Focusing on the differences of the examined classification approaches, it seems that cost-insensitive Logistic Regression is, by far, the best option for applications focusing on precision, achieving values between 0.85 and 0.97 for all datasets. However, this is achieved by allowing very significant losses in recall and F1 (values below 0.27 and 0.41 for all datasets, respectively). On the other hand, cost-sensitive Random Forest and Decision Tree classifiers seem to be the best options when recall and F1 are more important (albeit their losses in precision are significant).

4 RELATED WORK

The vast majority of works that attempt to estimate the expected impact of scientific articles focus on predicting the exact number of citations each article will receive in a given future period, a problem know as *Citation Count Prediction* (CCP). Most of these works incorporate a wide range of features based on the article's content, novelty, author list, venue, topic, citations, reviews, to name only a few. The corresponding predicting models are based

on various regression models like Linear Regression [22, 24], k-NN [22], SVR [10, 14, 22, 24], Gaussian Process Regression [21], CART Model [21, 22], ZINB Regression [4], or various types of neural networks [1, 11–13, 20, 24]. In most works, one or more regression models are tested on the complete data set, with the notable exception of [10], which first attempts to identify the current citation trend of each article (e.g., early burst, no burst, late burst, etc) and then applies a different model for each case. As it was elaborated in Section 2.2, CCP is a very difficult problem and there are many, not easily quantified factors that can significantly affect the performance of such approaches. Also, such approaches rely on article metadata that are difficult to collect and that they should be undergo complex to implement and time-consuming processing (see also Section 2.3).

In another line of work, based on the fact that co-authorship and citation-based features seemed to be effective for earlier approaches, the authors of [17] follow a link-prediction-inspired approach to solve CCP. They also investigate the effectiveness of their approach in a relevant classification problem based on a set of arbitrarily determined classes. However, training their approach requires a heavy pattern mining analysis of the underlying citation network and also considers author- and venue-based features, which face the already discussed issues. It should be noted that there are also some link prediction approaches that aim to reveal missing citations between a set of articles (e.g.,[23]), these approaches are irrelevant to the problem of impact prediction though. Furthermore, in [18] an impact-based classification problem is studied, but the features of the proposed approach rely on difficult to collect article metadata (e.g., information about academic and funding organizations). As a result, this approach cannot be easily used in practice. Finally, there are methods that attempt to estimate the rank of articles based on their expected impact. A thorough survey and experimental study of such methods can be found in [7]. This problem is easier than CCP, since only the partial ordering of the articles according to their expected impact should be estimated, but it is still more difficult than the problem we focus on.

5 CONCLUSION

In this work, we propose a simplified approach that can significantly simplify the work of researchers and developers working on applications that rely on the prediction of the expected impact of scientific articles. The proposed approach is based on classifying the articles in two categories ('impactful' / 'impactless') based on a set of features that can be calculated using a minimal set of article metadata. Furthermore, we experimentally evaluated this approach using various well-established classifiers showing that the results are more than adequate. The aforementioned experiments have been performed with caution taking into account the imbalanced nature of the classification problem at hand.

In the future, we plan to further investigate the imbalanced nature of the problem by examining other approaches like methods that perform over-sampling of the minority class, others that perform under-sampling of the majority class, or methods combining these two approaches (e.g., SMOTEEN). Additionally, we plan to examine a wider range of parameters for the examined approaches, for instance, examining a range of custom weights for cost-sensitive approaches. Finally, we plan to take full advantage of the Head/Tail Breaks approach to study a non-binary version of the classification problem.

ACKNOWLEDGMENTS

We acknowledge support of this work by the project "Moving from Big Data Management to Data Science" (MIS 5002437/3) which is implemented under the Action"Reinforcement of the Research and Innovation Infrastructur", funded by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

REFERENCES

[1] A. Abrishami and S. Aliakbary. 2019. Predicting citation counts based on deep neural network learning techniques. *Journal of Informetrics* 13, 2 (2019), 485–499.
[2] A. Barabási et al. 2016. *Network science.* Cambridge university press.
[3] J. Bollen, H. Van de Sompel, A. Hagberg, and R. Chute. 2009. A principal component analysis of 39 scientific impact measures. *PloS one* 4, 6 (2009), e6022.
[4] F. Didegah and M. Thelwall. 2013. Determinants of research citation impact in nanoscience and nanotechnology. *Journal of the American Society for Information Science and Technology* 64, 5 (2013), 1055–1064.
[5] H. He and Y. Ma. 2013. *Imbalanced learning: foundations, algorithms, and applications.* John Wiley & Sons.
[6] M. Jaradeh, A. Oelen, K. Farfar, M. Prinz, J. D'Souza, G. Kismihók, M. Stocker, and S. Auer. 2019. Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge. In *Proc. of K-CAP.*
[7] I. Kanellos, T. Vergoulis, D. Sacharidis, T. Dalamagas, and Y. Vassiliou. 2019. Impact-Based Ranking of Scientific Publications: A Survey and Experimental Evaluation. *IEEE TKDE* (2019).
[8] I. Kanellos, T. Vergoulis, D. Sacharidis, T. Dalamagas, and Y. Vassiliou. 2020. Ranking Papers by their Short-Term Scientific Impact. *arXiv preprint arXiv:2006.00951* (2020).
[9] P. Larsen and M. von Ins. 2010. The Rate of Growth in Scientific Publication and the Decline in Coverage Provided by Science Citation Index. *Scientometrics* 84, 3 (2010), 575–603.
[10] C. Li, Y. Lin, R. Yan, and M. Yeh. 2015. Trend-Based Citation Count Prediction for Research Articles. In *PAKDD.*
[11] M. Li, J. Xu, B. Ge, J. Liu, J. Jiang, and Q. Zhao. 2019. A Deep Learning Methodology for Citation Count Prediction with Large-scale Biblio-Features. *IEEE SMC* (2019), 1172–1176.
[12] S. Li, W. Zhao, E. Yin, and J. Wen. 2019. A Neural Citation Count Prediction Model based on Peer Review Text. In *EMNLP/IJCNLP.*
[13] L. Liu, D. Yu, D. Wang, and F. Fukumoto. 2020. Citation Count Prediction Based on Neural Hawkes Model. *IEICE Transactions on Information and Systems* (2020), 2379–2388.
[14] A. Livne, E. Adar, J. Teevan, and S. Dumais. 2013. Predicting citation counts using text and graph mining. In *Proc. of CompSci.*
[15] P. Manghi, C. Atzori, A. Bardi, J. Shirrwagen, H. Dimitropoulos, S. La Bruzzo, I. Foufoulas, A. Löhden, A. Bäcker, A. Mannocci, M. Horst, M. Baglioni, A. Czerniak, K. Kiatropoulou, A. Kokogiannaki, M. De Bonis, M. Artini, E. Ottonello, A. Lempesis, L. Nielsen, A. Ioannidis, C. Bigarella, and F. Summan. 2019. *OpenAIRE Research Graph Dump.* https://doi.org/10.5281/zenodo.3516918
[16] 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. 2011. Scikit-learn: Machine Learning in Python. *JMLR* 12 (2011), 2825–2830.
[17] N. Pobiedina and R. Ichise. 2016. Citation count prediction as a link prediction problem. *Applied Intelligence* 44, 2 (2016), 252–268.
[18] Z. Su. 2020. Prediction of future citation count with machine learning and neural network. In *IPEC.* IEEE, 101–104.
[19] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. 2008. ArnetMiner: Extraction and Mining of Academic Social Networks. In *KDD'08.* 990–998.
[20] J. Wen, L. Wu, and J. Chai. 2020. Paper Citation Count Prediction Based on Recurrent Neural Network with Gated Recurrent Unit. *IEEE ICEIEC* (2020), 303–306.
[21] R. Yan, C. Huang, J. Tang, Y. Zhang, and X. Li. 2012. To better stand on the shoulder of giants. In *Proc. of ACM/IEEE-CS JCDL.* 51–60.
[22] R. Yan, J. Tang, X. Liu, D. Shan, and X. Li. 2011. Citation count prediction: learning to estimate future citations for literature. In *Proc. of CIKM.* 1247–1252.
[23] X. Yu, Q. Gu, M. Zhou, and J. Han. 2012. Citation Prediction in Heterogeneous Bibliographic Networks. In *SDM.*
[24] X. Zhu and Z. Ban. 2018. Citation Count Prediction Based on Academic Network Features. *IEEE AINA* (2018), 534–541.

A USED PARAMETER CONFIGURATIONS

Tables 5 & 6 summarize the configuration for each used approach. The names of the parameters are based on the input parameters of the corresponding Scikit-learn functions. Omitted input parameters were not configured (their default values had been selected).

Classifier	Configuration for $y = 3$	Configuration for $y = 5$
LR_{prec}	'max_iter': 200, 'solver': 'sag'	'max_iter': 160, 'solver': 'sag'
LR_{rec}	'max_iter': 80, 'solver': 'sag'	'max_iter': 80, 'solver': 'sag'
LR_{f1}	'max_iter': 180, 'solver': 'sag'	'max_iter': 240, 'solver': 'sag'
cLR_{prec}	'max_iter': 100, 'solver': 'sag'	'max_iter': 60, 'solver': 'sag'
cLR_{rec}	'max_iter': 120, 'solver': 'sag'	'max_iter': 140, 'solver': 'sag'
cLR_{f1}	'max_iter': 180, 'solver': 'sag'	'max_iter': 140, 'solver': 'sag'
DT_{prec}	'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 2	'max_depth': 4, 'min_samples_leaf': 1, 'min_samples_split': 2
DT_{rec}	'max_depth': 1, 'min_samples_leaf': 1, 'min_samples_split': 2	'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 2
DT_{f1}	'max_depth': 1, 'min_samples_leaf': 1, 'min_samples_split': 2	'max_depth': 8, 'min_samples_leaf': 10, 'min_samples_split': 200
cDT_{prec}	'max_depth': 1, 'min_samples_leaf': 1, 'min_samples_split': 2	'max_depth': 1, 'min_samples_leaf': 1, 'min_samples_split': 2
cDT_{rec}	'max_depth': 2, 'min_samples_leaf': 1, 'min_samples_split': 2	'max_depth': 2, 'min_samples_leaf': 1, 'min_samples_split': 2
cDT_{f1}	'max_depth': 7, 'min_samples_leaf': 4, 'min_samples_split': 20	'max_depth': 7, 'min_samples_leaf': 4, 'min_samples_split': 50
RF_{prec}	'criterion': 'gini', 'max_depth': 1, 'max_features': 'log2', 'n_estimators': 200	'criterion': 'gini', 'max_depth': 1, 'max_features': 'log2', 'n_estimators': 200
RF_{rec}	'criterion': 'gini', 'max_depth': 10, 'max_features': 'log2', 'n_estimators': 300	'criterion': 'gini', 'max_depth': 10, 'max_features': 'sqrt', 'n_estimators': 300
RF_{f1}	'criterion': 'entropy', 'max_depth': 10, 'max_features': 'sqrt', 'n_estimators': 200	'criterion': 'entropy', 'max_depth': 10, 'max_features': 'sqrt', 'n_estimators': 300
cRF_{prec}	'criterion': 'entropy', 'max_depth': 1, 'max_features': 'log2', 'n_estimators': 150	'criterion': 'entropy', 'max_depth': 1, 'max_features': 'log2', 'n_estimators': 100
cRF_{rec}	'criterion': 'gini', 'max_depth': 5, 'max_features': 'sqrt', 'n_estimators': 150	'criterion': 'entropy', 'max_depth': 5, 'max_features': 'log2', 'n_estimators': 100
cRF_{f1}	'criterion': 'entropy', 'max_depth': 10, 'max_features': 'log2', 'n_estimators': 150	'criterion': 'gini', 'max_depth': 5, 'max_features': 'sqrt', 'n_estimators': 300

Table 5: Parameter configurations for PMC.

Classifier	Configuration for $y = 3$	Configuration for $y = 5$
LR_{prec}	'max_iter': 80, 'solver': 'sag'	'max_iter': 100, 'solver': 'sag'
LR_{rec}	'max_iter': 80, 'solver': 'sag'	'max_iter': 140, 'solver': 'sag'
LR_{f1}	'max_iter': 220, 'solver': 'saga'	'max_iter': 220, 'solver': 'sag'
cLR_{prec}	'max_iter': 200, 'solver': 'sag'	'max_iter': 180, 'solver': 'sag'
cLR_{rec}	'max_iter': 140, 'solver': 'sag'	'max_iter': 160, 'solver': 'sag'
cLR_{f1}	'max_iter': 100, 'solver': 'sag'	'max_iter': 60, 'solver': 'newton-cg'
DT_{prec}	'max_depth': 6, 'min_samples_leaf': 1, 'min_samples_split': 2	'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 2
DT_{rec}	'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 2	'max_depth': 1, 'min_samples_leaf': 1, 'min_samples_split': 2
DT_{f1}	'max_depth': 3, 'min_samples_leaf': 1, 'min_samples_split': 2	'max_depth': 4, 'min_samples_leaf': 1, 'min_samples_split': 2
cDT_{prec}	'max_depth': 14, 'min_samples_leaf': 10, 'min_samples_split': 2	'max_depth': 4, 'min_samples_leaf': 1, 'min_samples_split': 2
cDT_{rec}	'max_depth': 2, 'min_samples_leaf': 1, 'min_samples_split': 2	'max_depth': 2, 'min_samples_leaf': 1, 'min_samples_split': 2
cDT_{f1}	'max_depth': 11, 'min_samples_leaf': 10, 'min_samples_split': 200	'max_depth': 4, 'min_samples_leaf': 1, 'min_samples_split': 2
RF_{prec}	'criterion': 'entropy', 'max_depth': 1, 'max_features': 'log2', 'n_estimators': 150	'criterion': 'gini', 'max_depth': 5, 'max_features': 'sqrt', 'n_estimators': 100
RF_{rec}	'criterion': 'entropy', 'max_depth': 1, 'max_features': 'log2', 'n_estimators': 150	'criterion': 'entropy', 'max_depth': 1, 'max_features': 'log2', 'n_estimators': 150
RF_{f1}	'criterion': 'gini', 'max_depth': 5, 'max_features': 'log2', 'n_estimators': 100	'criterion': 'entropy', 'max_depth': 10, 'max_features': 'sqrt', 'n_estimators': 250
cRF_{prec}	'criterion': 'entropy', 'max_depth': 1, 'max_features': 'log2', 'n_estimators': 250	'criterion': 'entropy', 'max_depth': 1, 'max_features': 'log2', 'n_estimators': 100
cRF_{rec}	'criterion': 'gini', 'max_depth': 5, 'max_features': 'log2', 'n_estimators': 100	'criterion': 'gini', 'max_depth': 1, 'max_features': 'log2', 'n_estimators': 150
cRF_{f1}	'criterion': 'entropy', 'max_depth': 10, 'max_features': 'log2', 'n_estimators': 150	'criterion': 'entropy', 'max_depth': 10, 'max_features': 'sqrt', 'n_estimators': 150

Table 6: Parameter configurations for DBLP.