



**Department of Electrical, Computer
& Biomedical Engineering**
Faculty of Engineering & Architectural Science

Prediction Publication Performance

by

Muhammad Karim, Erfan Kashani, Fatima Tafazzoli Shadpour, and Lubaba Tasnim

Computer/Electrical Engineering Capstone Design Project

Ryerson University, 2021

Acknowledgements

We would like to express our very great appreciation to Dr. Ebrahim Bagheri for his valuable and constructive suggestions during the planning and development of this project. His willingness to give his time so generously has been very much appreciated. We have especially valued the clarity and detail of his feedback on drafts of the design as well as providing resources during the implementation.

We would also like to thank the Department of Electrical, Computer and Biomedical Engineering and the course coordinators for the resources provided.

Certification of Authorship

We hereby certify that the work completed in the duration of the final engineering design report has not previously been submitted for a degree. Furthermore, this document has not been a part of the requirements for a degree except as fully acknowledged within the text. We also certify that the content of this report has been written by the member of this capstone group. Any assistance and guidance we have received in the research process, as well as in the preparation of the report itself has been acknowledged. Lastly, we certify that the information source and literature indicated within the report has been authenticated.

Signature of Candidates





| Name | Student ID | Signature |
|---------------------------|------------|---|
| Erfan Kashani | 500757016 |  |
| Fatima Tafazzoli Shadpour | 500767196 |  |
| Lubaba Tasnim | 500758704 |  |
| Muhammad Karim | 500774436 |  |

Table of Contents

| | |
|---|----|
| Acknowledgements | 2 |
| Certification of Authorship | 3 |
| Abstract | 5 |
| 1. Introduction & Background | 6 |
| 2. Objectives | 8 |
| 3. Theory and Design | 9 |
| 3.1 Theory | 9 |
| 3.1.1 Random Forest Regression Model | 9 |
| 3.2. Design | 11 |
| 3.2.1. Data Selection | 12 |
| 3.2.2. Dataset Retrieval | 12 |
| 3.2.3. Dataset Cleaning and Processing | 12 |
| 3.2.4. Feature Extraction | 12 |
| 3.2.5. Model Selection | 14 |
| 3.2.6. Model Prediction | 14 |
| 3.2.7. Model Evaluation | 15 |
| 3.2.8. Feature Engineering | 15 |
| 3.2.9. Visualization | 16 |
| 4. Alternative Designs | 17 |
| 4.1. Alternative Feature Design | 17 |
| 4.2. Alternative Model Design | 18 |
| 4.2.1. Gradient Boosting | 18 |
| 4.2.2. Classification and Regression Tree | 19 |
| 5. Material/Component list | 20 |
| 6. Measurement and Testing Procedures | 21 |
| 7. Performance Measurement Results | 22 |
| 8. Analysis of Performance | 31 |
| 9. Conclusions | 33 |
| References | 34 |
| Appendices | 35 |
| Appendix A.1 | 35 |
| Appendix A.2 | 37 |

Abstract

In the rise of the current rate of publication, creating and identifying influential papers are challenging tasks for researchers. The objective of this project is to predict the impact of an academic paper in future using the citations metric. The team performs an extensive study into the current methodologies for this machine learning problem. Consequently, the project arrives at multiple ensemble learning algorithms to perform the task. This study utilizes the famous Aminer dataset for training and testing purposes. We examine sets of fundamental features related to papers, authors, and venues to discover the highly cited papers' patterns. After an iterative process of data analyzing and processing, 10 independent features are selected to represent the sample set. Then, a model is generated to predict the paper's citation within one to ten years of publication. We considered many well-known machine learning algorithms and compared the results based on their performance on the Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The three candidate models include Random Forest Regression, Gradient Boosting, and Classification and Regression Tree (CART) model. Each model is enhanced using hyperparameter optimization, and feature engineering. The Random Forest Regressor was chosen as the best candidate due to its robustness, flexibility, and ability to limit overfitting without increasing error for the dataset. The ultimate Random Forest Regression model was able to predict the one year impact of a paper with 1.168 MAE, 43.388 MSE, and 6.59 RMSE values. The results were compared with the benchmark models which include: Linear Regression, Support Vector Machine, and K-Means clustering. The suggested model's result shows improvements on the benchmark models used for prediction. Finally, the studies suggest a correlation between the paper's citation and its author's social and academic reputation.

Keywords: Citation pattern prediction, Machine learning, Influence prediction, Regression model, Data engineering, Scientific impact

1. Introduction & Background

In recent years, there has been a substantial amount of research done regarding the impact of academic papers. Nevertheless, individuals have limited time and do not want to waste time reading papers of no significance. Many would be more interested in reading papers that have a high impact in their fields so that they can get ahead of their peers in contributing to an emerging field. For papers that have already been published for more than five years, we can evaluate the citation impact of the paper using its citation count. However, for papers that have only been published recently, it is almost impossible to decide whether a paper would make a high citation impact. The citation count is a significant indicator considering it is used for measuring the impact of a paper, and it has been used as the basis for many other metrics. Furthermore, the prediction of publication performance is one of many modern-day applications of machine learning and data science. The analysis of data has credible merits such as identifying trends, simplifying data entry tasks, and improving the efficiency of machine training without human interaction. This process allows researchers to combine analytics, statistics, and machine learning to make predictions. Additionally, large datasets can be utilized to make data-driven decisions based on the findings. As a result, organizations will be able to develop cutting-edge competitive technology and create accurate forecasts. Moreover, big data analytics help improve a company's engineering infrastructure, efficiency, skills and marketing objectives. [1]

The team explored the topic of Prediction Publication Performance because of its unique and versatile applications. There are many studies conducted in recent years regarding predictive modeling. Yan (2012) discussed the merits of Gaussian Process Regression (GPR). The model utilizes Gaussian distribution to predict errors using the central limit theorem, the mean vector and covariance matrix (known as a kernel). The kernel is then used to indicate the influence between points within the distribution. It can also determine the mean and confidence interval which models the posterior probability of each sample [3]. It observed how the citation count of a paper can be influenced by features such as the content of the paper, the expertise of the authors, and the impact of the venue. The prediction performance is evaluated with the coefficient of determination, also known as R-squared. Results show that the authors had a bias when making citations. Furthermore, features relating to the expertise of the author and the influence of the venue have a significant impact when considering which papers should be cited.

Artificial Neural Networks (ANN), specifically the Backpropagation Neural Network (BPNN) models, create a network architecture of input, hidden and output layers. The hidden layers take key information from the input nodes and pass it to the next layer. This iteration is done to pass the important data through the entire network, until the final prediction at the output layer is determined. An activation function is used to capture the nonlinear relationships and provide an output. The model can be optimized through the gradient descent [4]. This model was used by Ruan (2020) and examined how BPNNs can be used for the task of prediction. One of the main advantages of the model was its ability to automatically learn features through nonlinear transformations. The model was able to successfully predict the impact of a paper for the first and second year after publication. It achieves a better performance than the baseline models. It also reveals that the high accuracy of the model will allow this method to be used in potential literature retrieval and future scientific evaluations.

Lastly, Yan (2011), proposes the Classification and Regression Tree (CART) model as the ideal solution for the prediction task. The CART model implements a greedy optimization method which divides features based on condition. It utilizes criteria such as the Gini Impurity to split nodes and can support the prediction for continuous variables. Additionally, cross-validation was applied to prevent overfitting, and estimate how the model will perform. It is once again observed that authors have a bias when selecting which papers should be cited. The coefficient of determination is once again utilized to evaluate prediction performance. In particular, the author rank and venue rank have a significant impact on this decision.

2. Objectives

The problem of citation prediction is one of the most difficult estimates to evaluate due to the fact that there is not a set of patterns or formulas to calculate it. Moreover, with many researchers working and writing research papers, many papers are being published every day with different impacts. Thus, taking the basic information of a paper and predicting the amount of impact it will have seems significantly challenging. In this paper our aim is to create a machine learning program to take on this problem. The main task for the model is to take features driven from the attributes of the papers and return the number of citations annually up to ten years after publication. Furthermore, to achieve this, it is essential to design and extract features using only the basic information of the paper. These features not only require to account for the characteristics of the paper; but also the influence of the authors and the publication venue of the paper. Finally it is important to compare the model to standard and traditional baseline models to evaluate validity and reliability of it.

3. Theory and Design

3.1 Theory

3.1.1 Random Forest Regression Model

The Random Forest model is a supervised learning algorithm that is capable of performing classification and regression tasks [7]. This model takes randomized samples from the dataset to form a large number of decision trees, this ensemble is referred to as a forest. A decision tree consists of nodes (a condition that helps classify data), which are connected to each other through branches (the possibilities that a decision can lead to) [8]. An example of a decision tree is illustrated below in Figure 3.1.1. Each tree consists of the following types of nodes:

1. Root Node: Highest node in the decision tree, which does not have a predecessor but does have successors.
2. Internal (Non-leaf) Node: A node located between the root and leaves, which have one or more successors.
3. Leaf Node: Does not have any successors, and contains the results of a sequence of decisions.

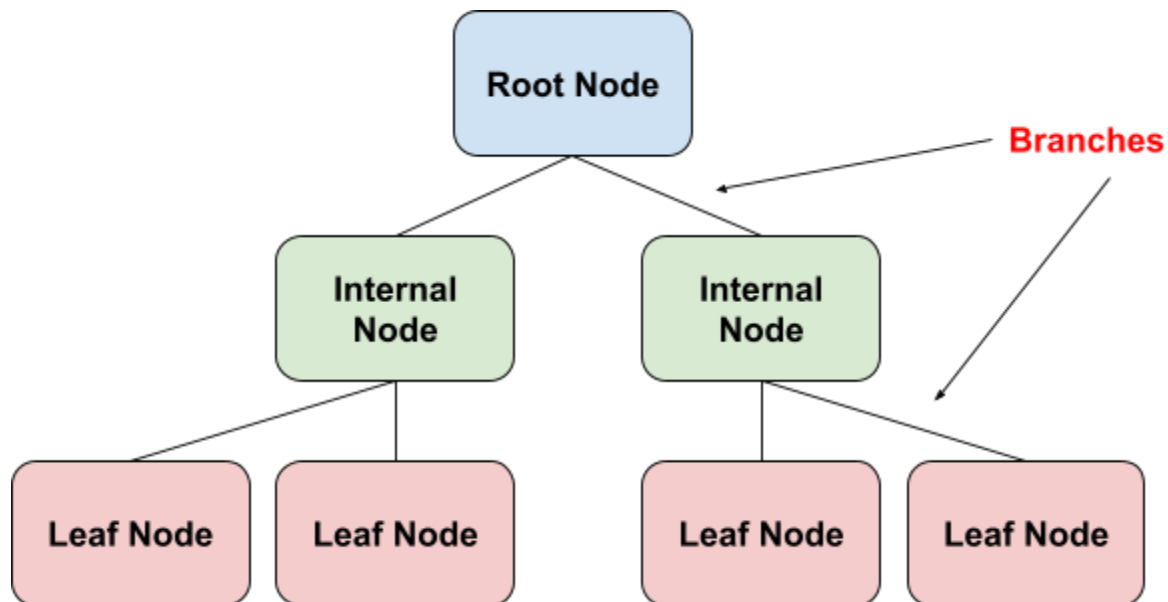


Figure 3.1.1. Decision Tree Structure

Each tree within the forest will produce a prediction based on the collection of data each model contains. The class which appears the most frequently will be the prediction made by the model. There are many algorithms that can be used to determine these predictions.

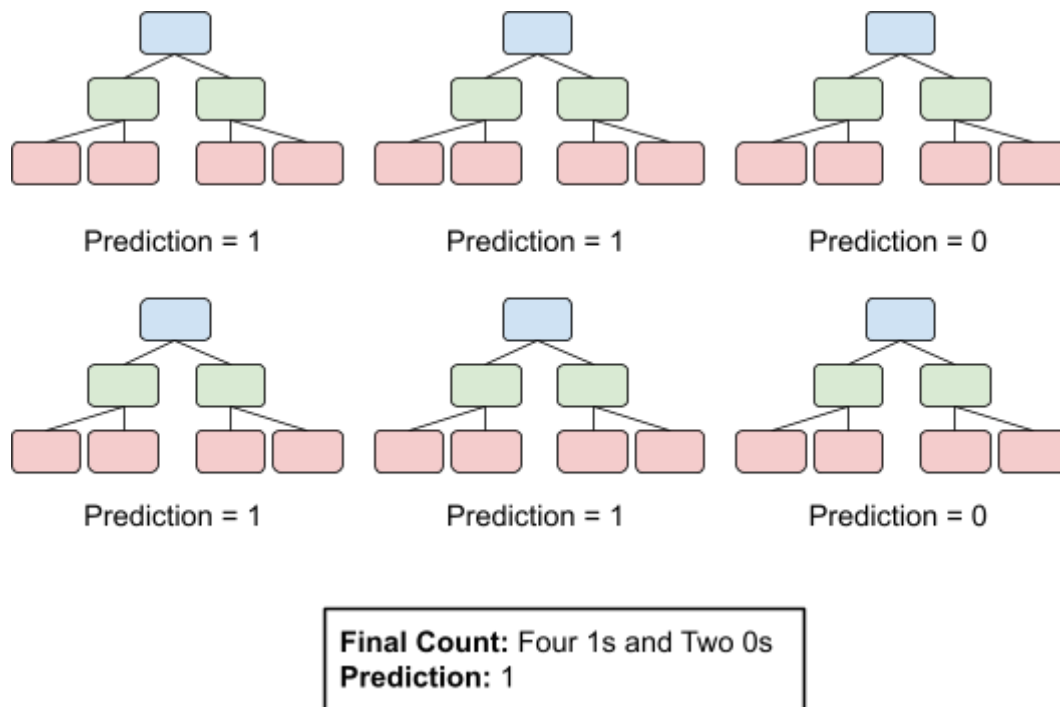


Figure 3.1.2. Random Forest Model Prediction Example

The random selection of data allows a chance of a lower correlation between decision trees. This is advantageous for the purpose of decreasing the model variance because a large collection of uncorrelated decision trees has a higher probability of outperforming a single decision tree model [8]. An increase in the correlation between trees also results in a higher rate of error. If an individual tree has a lower rate of error it is a stronger classifier. It is important to consider this because a group of learning models will prevent each other from committing the same errors, and increase the precision of the overall results. [8]

To ensure that the forest consists of a diverse set of decision trees, two methods can be applied. The first method will allow each tree to randomly select samples from the dataset, resulting in different trees. This method is known as “bagging”. This is important because the Random Forest algorithm is sensitive to the data trained on, and minor adjustments may lead to significant variance in structures. The bootstrapping technique will select a random size of samples from the training set where the same sample can be taken more than once. For example, if the dataset contains the values $\{1, 2, 3, 4, 5\}$, then one of the possible decision trees could be $\{1, 3, 4, 5, 5\}$. The second method which can be applied here is enlisting feature randomness. A tree within the Random Forest can only select samples from a random subset of the features. Thus allowing a higher variation between decision tree models. [8]

When implementing Random Forest to solve regression problems, the mean squared error (MSE) and mean absolute error are used to represent the data at each node. Equation 3.1.1 below shows the MSE equation. Here, N represents the total number of data points, y_i represents the value observed, and \hat{y}_i represents the predicted value at point i .

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (3.1.1)$$

3.2. Design

The Citation Prediction Performance considers Random Forest Regression as the primary model. The following flowchart (Figure 3.2.1) illustrates the stages of project implementation. This includes dataset selection, data retrieval, feature extraction, model selection, training, and model evaluation. This section will further elaborate on the design and implementation process.

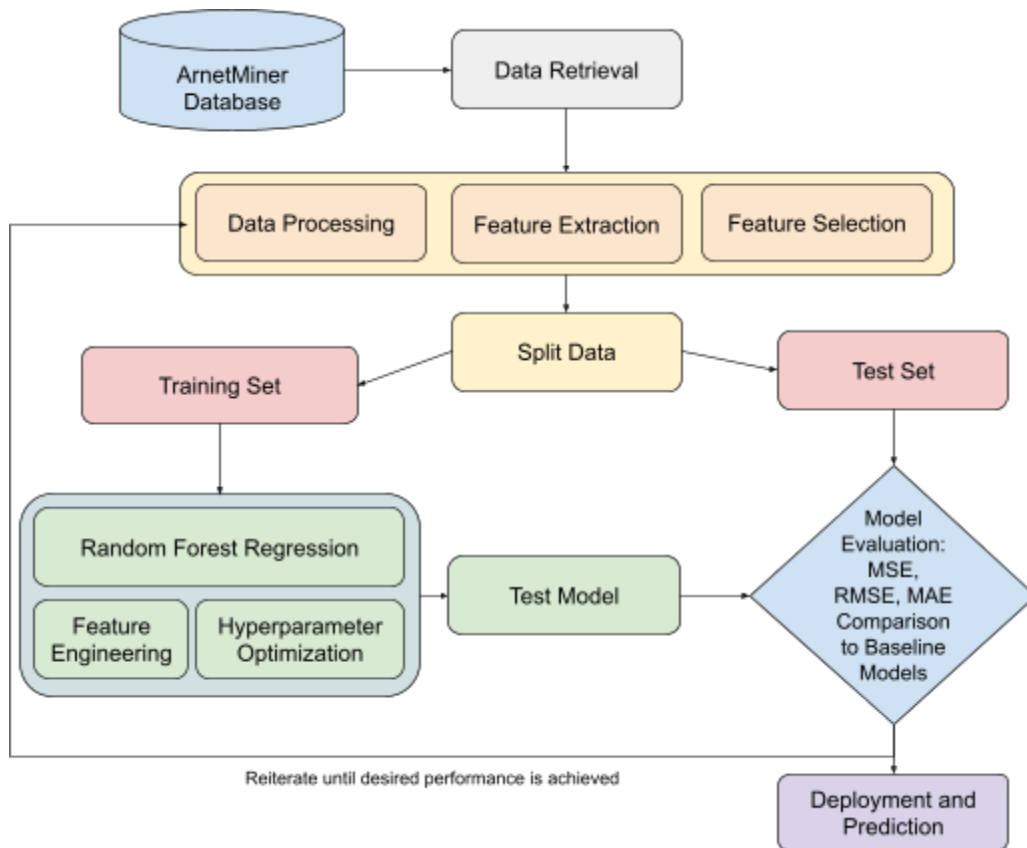


Figure 3.2.1. Flowchart of the study

3.2.1. Data Selection

After considering the pros and cons of multiple datasets, the research team decided to utilize the twelfth version of the ArnetMiner (AMiner) database, which was established in April 2020 [9]. Aminer was selected due to its versatility, use of clustering with network related information, and study of influence in the citation network to find the influential papers. Citation network dataset consists of 4,894,081 research papers, 45,564,149 citation relationships, and is stored with the JSON format. This version includes features relating to the author, paper, venue and fields of study. The features are being selected and extracted on a case-to-case basis, due to the variance in the computational complexity of each feature. Additionally, the correlation of each feature and its effects on the model will be considered.

3.2.2. Dataset Retrieval

The dataset can be parsed from a JSON file for further analysis and manipulating the data. The process of parsing the data begins with defining the data file path, and using yield to load the JSON file through a loop to prevent memory overload. The yield function is used to extract the data line-by-line. Due to the large size of the file and limited processing power, the data is loaded and stored based on the year. If the fetched data belongs to the particular year, it is appended to the paper list array and all of the information for that year is saved and printed. Once this is complete, the data frame is converted into binary form using the Pickle serialization. This method allows the team to easily import sections of data and perform future calculations.

3.2.3. Dataset Cleaning and Processing

Once the data is retrieved, each attribute is analysed to determine its data type and help account for missing values and outliers. Each column in the database is divided according to the author, paper, and venue categories. Any features which are not relevant to this design project were dropped. The cleaning procedure begins by checking for the null values within each attribute, and eliminating any missing entries. All of the attributes were casted to the appropriate data type. A box plot is generated to show the citation trends inside of the dataset. Moreover, the data was observed to be skewed and showed signs of nonlinearity. This allows the team to organize the data, and perform important citation evaluation to improve the upcoming model.

3.2.4. Feature Extraction

As mentioned previously, the features selected for the design are done based on a case-by-case basis. The team has considered a total of ten features, which will encompass the paper, author, and venue features. The features being utilized in this feature set was determined based on a series of feature engineering tasks, which will be discussed later in this chapter. Below, Table 3.2.1 states all the features being included in this design project, and their descriptions.

Table 3.2.1. The model's features and their description

| Feature name | Description |
|--------------|---|
| Diversity | The number of topics used in the paper |
| Venue Rank | The average citation is calculated, and the venue is ranked based on order |
| Venue MPI | Maximum citation count of a venue |
| Venue TPI | Total citation count of a venue |
| Productivity | The number of papers an author has published |
| H-index | H-index of the authors before the current article |
| Author Rank | The average citation is calculated, and the author is ranked based on the order |
| Author MPI | Maximum citation count of an author |
| Author TPI | Total citation count of an author |
| Versatility | The number of distinct topics studied by the author |

This model also utilises features from the original dataset such as ID, authors, title, year, references, field of study, and venue, these are utilized to extract the paper ids from a given year. In this case, the year was 2010 to enable capturing the 10 year influence. The most significant changes in the updated feature set are the calculated values for the number of citations of a number of years after publication. In addition to this the number of citations (n_citation) is compared to the calculated number of citations (citation), to observe the difference. This implementation process of calculating the citation values begins with dropping missing values from the reference column of the original dataset. Then, the features which are in the form of lists are split using the explode function. This will duplicate the singular values, and separate the values within the list into rows (illustrated in Figure 3.2.2)

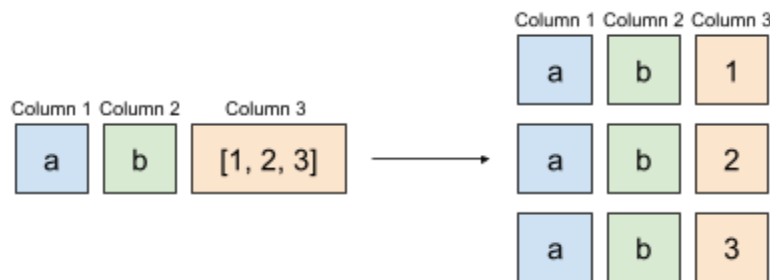


Figure 3.2.2. Value Explode Process

Once this is complete, the process of grouping takes place to categorize values based on the commonality. This will take a given value, and all of the rows which contain that value will be grouped together. Values within the same column will be grouped together as a list. An example of this is shown in Figure 3.2.3 below:

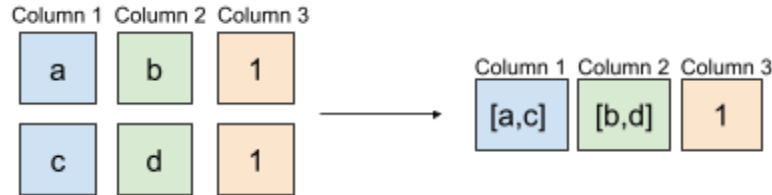


Figure 3.2.3. Value Grouping Process

The value in the reference column is the name of the paper, and the values in the other columns are the ones that cited the paper. The id column is a list showing the papers which reference a given research paper. The length of that list will be the number of citations. The difference between the year of publishing and year of reference will determine how many years after publication a paper is cited (ex. if the year of publication is 2005 and the year of reference is 2006, then the difference will be 1). An iteration is done to check the year of reference, and length of the number of citations is used to determine the values of citation in one to five years, seven years, ten years of publication and the total number of citations. These features will be discussed further later in this section.

3.2.5. Model Selection

The model design will be implemented through the Random Forest Regression. Once the features are analysed, any missing data will be dropped, and all remaining features shall be cleaned further. Next, the dataset will be divided between a training and testing set. The training set consists of papers published until the year 2009 while the testing set is limited to the papers published in 2010. By observing the trends of the dataset, the year 2010 and a nonlinear model were chosen to capture the 10-year citation influence of papers similar to the research presented in Yan et al., 2012 and Ruan et al., 2020.

3.2.6. Model Prediction

The model prediction process begins with adding the database for the features extracted for a set of years, and another with information regarding the paper ids from one year. The data is split for training and testing. The model uses the training set for discovering the data trends and the testing set is used to calculate the model performance. The RandomForestRegressor consists of a set of hyperparameters such as number of trees in the forest, number of samples required to split an internal node, and maximum depth of the tree. These values are utilized for parameter optimization, and to improve the overall model. This estimator is used to train the model for the number of post-publication years. Table 3.2.2 demonstrates the final hyperparameters used for this model.

Table 3.2.2: Hyperparameters for RandomForestRegressor model

| n_estimators | Max Iteration | Min Sample Split | Min Samples Leaf | Max Features | Max Depth | Bootstrap | Random State |
|--------------|---------------|------------------|------------------|--------------|-----------|-----------|--------------|
| 32 | 100 | 2 | 2 | auto | 60 | True | 0 |

3.2.7. Model Evaluation

The process of model evaluation will compare the authentic citation values against the predicted citation values from the testing set. There are three evaluation methods being considered for part of the procedure, the Mean Squared Error (MSE), the Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). Since the importance and influence of a feature will vary based on the model, a detailed assessment of the contributing factors will be required following the model evaluation. Once this stage is complete, feature analysis will be performed to observe the importance of each feature's importance within the overall performance of the model. In addition to this, histograms were generated to visualize how closely the prediction is compared to the train and test data for each year. It was observed that the predicted value closely follows the same trend at the train and test data.

3.2.8. Feature Engineering

Feature engineering closely examines the importance of each feature, in terms of what impact each feature has on the end results. Analyzing the impact of features is a crucial step in the hyperparameter tuning process which will help to improve the existing model. The goal of this is to alter features in a manner to reduce noise and model complexity while increasing the performance. Graphs are generated to determine which features are most important in reference to the improvement of the model. Based on the resulting graphs the top 9 features which have the most impact account for 95% of the influence on the model are determined. In addition to this, the model accuracy was tested by removing the feature with the least amount of impact to reduce noise and observe possible changes in accuracy.

Hyperparameter tuning is necessary to make adjustments to the model and optimizes its overall performance. In the Random Forest model, the hyperparameters mainly control the settings of the underlying decision trees. The decisions are being made by splitting nodes between the root level to the leaf level(s). While tuning does usually involve trial-and-error, it is possible to utilize different combinations of features to evaluate model performance.

There are a number of hyperparameters which can be included in a Random Forest Regressor. Prior documentation from the scikit learn library indicates that the number of trees (`n_estimators`) is the most important hyperparameter. This model considers the following:

1. `n_estimators`: the number of decision trees within a forest
2. `max_features`: the number of features being considered for the best split
3. `max_depth`: the maximum depth of the tree
4. `min_samples_split`: the minimum number of samples needed to split an internal (non-leaf) node
5. `min_samples_leaf`: the minimum number of samples at a leaf node
6. `bootstrap`: whether bootstrap sampling (test or metric that uses random sampling with replacement) will be implemented.

A parameter grid is developed using the Randomized Search CV. This contains a range of values for each parameter, which can be sampled during fitting. The random search will look through the values. During each iteration the algorithm can choose different combinations through these hyperparameters.

Cross validation is a technique which allows one to ensure that the model is performing well on the training and test set, but also preventing overfitting. One particular method of this, is known as the K-Fold cross validation. This will split the training set into a number of subsets, known as folds. The model is fitted through iteration, where the training data of the fold is represented by $K-1$ and the validation fold is represented by K . In the case of this Random Forest model, $K = 3$. Meaning the data will be divided into three parts. It begins by training two-thirds of the data, and validating one-thirds of the data. In the first iteration, the first two folds are trained and the third is evaluated. In the second iteration, the first and third folds are trained and the second fold is evaluated. Lastly, in the final iteration, the second and third folds are trained and the first fold is evaluated. These 3 permutations are performed a total of 10 times. Each time, it uses a different model setting. Once this is complete, all of the models are compared and the best one is observed for training and evaluation.

3.2.9. Visualization

Streamlit library was utilized to create an interactive web application which presents the results. The application demonstrates the prediction pipeline of a paper including the steps taken during feature extraction, predicting the citation results, and comparing them against the true values in the dataset. Moreover, the result page presents the overall research findings using graphs and tables.

4. Alternative Designs

To implement the optimal model, a series of iterative designs and evaluations were performed. During this process, the research team constructed alternatives designs and compared their constraints and performance to find the best solution. The alternative designs are further divided into feature designs and model design which are discussed separately.

4.1. Alternative Feature Design

Initially, the team predicted that three major factors are influential for a research paper's citation count of a research paper. These features include: paper related, author related, and venue related information. Consequently, the research team arranged the following feature set from the dataset (Table 4.1.1).

Table 4.1.1: Initial feature design description

| Feature Name | Feature Type |
|--|-----------------|
| Document type | Paper feature |
| Paper length | Paper feature |
| Title length | Paper feature |
| Number of publications in the journal | Journal feature |
| Number of authors | Author feature |
| Productivity | Author feature |
| Previous citations | Author feature |
| Previous citations per article | Author feature |
| h-index | Author feature |
| Number of organizations | Venue feature |
| Total number of papers produced in the institute | Venue feature |
| Max H-index of Author in Organization Impact of faculty members (average H-index) | Venue feature |
| Number of References | Venue feature |
| Average Reference Age | Venue feature |

| | |
|--------------------------------|---------------|
| Self-citations of journals | Venue feature |
| Percentage of journal articles | Paper feature |
| Percentage of conference paper | Paper feature |
| Percentage of books | Paper feature |

Interestingly it was discovered that paper features were not significantly affecting the citations. Therefore, author and venue related features had much more correlation with the citation count. This conveys the idea that the author's social connections and reputation in the field of study plays an important role on their citation count.

Moreover, in the starting design alternatives, the research team used the `n_citation` column provided by the dataset as the ground truth for each observation. After analysis of the independent correlation between features with the dependent one (`n_citation`), the group concluded that the constraints of using this information for the correct prediction. The problem resides in the time-independent nature of the `n_citation` column. As a result, it was difficult to compare citation counts of two papers in the fixed time frame. For example, the `n_citation` of a paper published in 2010 reflects the 10 years citation effect while the `n_citation` of a paper published in the year 2000 shows the 20 years influence. Therefore, the citation count scope was limited to the references that have been made within the papers of the dataset. Using this approach, the features could be extracted for annually up to five years and seven and ten years' influence period after the publication.

4.2. Alternative Model Design

4.2.1. Gradient Boosting

Gradient Boosting Regression has gained popularity in recent years [10]. Unlike the Random Forest algorithms that create independent trees to predict the results, the Gradient Boosting algorithm constricts a series of shallow trees where each tree attempts to lower the errors from the previous one. The idea is to sequentially boost the performance by adding a new tree to the ensemble. The design utilized a special kind of Gradient Boosting algorithm called `HistGradientBoostingRegressor` from the `sklearn` library. The data was preprocessed into a training and testing set. Table 4.2.1 shows the hyperparameters used for this model:

Table 4.2.1: Hyperparameters for `HistGradientBoostingRegressor` model

| Learning rate | Max Iteration | Max Leaf Nodes | Min Sample Leaf |
|---------------|---------------|----------------|-----------------|
| 0.15 | 100 | 31 | 20 |

The model was trained on one to ten years datasets respectively. The evaluation showed competitive MAE, MSE, and RMSE values (shown in Appendices section), however; histograms indicated model overfitting toward higher citation papers and inaccurate prediction for papers with zero citations. It is important to note that the gradient boosting model is not computationally cost efficient and this resulted in difficulties during the hyperparameter tuning stage. The model parameters were optimized using the grid search method and cross validation techniques. The final model results were not able to outperform the Random Forest Regression model. Therefore, this model was chosen as an alternative design.

4.2.2. Classification and Regression Tree

Classification and regression tree (CART) was the second alternative model. The goal of this machine learning algorithm is using a decision tree to partition the feature set into smaller portions using conditions. The CART model uses a greedy optimization technique to perform partitioning. To start it assigns the Root node condition using the feature with the lowest impurity value. As the root node divides into smaller intermediate nodes, the size of the samples applied to each node decreases. By the end, the tree diverges into leaf nodes where a separation would not lower the impurity any further. The impurities were calculated using the Gini criterion [11]. The implemented procedure started by preprocessing the data for the one to ten years citation counts. Each training set was inputted into the CART model separately and the trained model was evaluated on the testing set. Table 4.2.2 shows the hyperparameters used for the CART model.

Table 4.2.2: Hyperparameters for CART model

| Criterion | Splitter | Max Depth | Min Sample Split | Min Sample Leaf | Random State |
|-----------|----------|-----------|------------------|-----------------|--------------|
| Gini | best | 25 | 4 | 2 | 0 |

After the evaluation step, the CART model showed overfitting results toward the papers with zero citation. This is due to the high variance of the CART model and requires further pruning techniques. Therefore, this model was not able to outperform the primary design.

5. Material/Component list

Components required for this project are as follows:

- ArnetMiner (AMiner) Citation V12 dataset [9]
 - The citation data is extracted from DBLP, ACM, and other sources.
 - Each paper is associated with abstract, authors, year, venue, and title.
 - This version includes the following attributes for each paper: id, title, name of authors (authors.name), organization of authors (authors.org), ids of authors (authors.id), ID of the published venue (venue.id), full name of the venue (venue.raw), year, name of the fields of study (fos.name), weights and effects of each field on the paper (fos.w), IDs of the papers referenced (references), total citations (n_citation), starting page (page_start), end page (page_end), type of document (doc_type), publisher, volume, issue, doi, text index of the abstract (indexed_abstract).
- Git software
 - Version control software used for tracking changes in any set of files
- Python Version 3.6+
- Anaconda
 - A distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment.
- Jupyter Notebook
 - A server-client application that allows editing and running notebook documents (documents which contain both computer code and rich text elements) via a web browser.
- Pandas library
 - A fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.
- Numpy library
 - A Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.
- Sklearn library
 - A free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.
- Streamlit library
 - Streamlit is an open-source app framework to create custom web apps for machine learning and data science projects.

6. Measurement and Testing Procedures

To evaluate our model and design, quantitative and graphical evaluations have been carried out. For the quantitative measurements, the Mean Squared Error, Root Mean Squared Error, Mean Absolute Error and R-Squared values have been calculated. Mean Squared Error, often used in regression machine learning models, is an evaluation metric. The MSE is the mean of the squared prediction errors (difference between the true value and the predicted value) over all instances [12]. The Root Mean Squared Error is the root square of the MSE value. RMSE is used to measure the quality of predictions and is the standard deviation of the residuals. That is to say, the RMSE demonstrates how spread out the residuals are [13]. in equations 6.1.1 to 6.1.4, y_i , \hat{y}_i and \bar{y} represents the predicted, true and average values respectively. Furthermore, N represents the number of samples in the test dataset.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (6.1.1)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (6.1.2)$$

Mean Absolute Error (MAE) represents the mean of the absolute values for the differences between the true values and the predicted values over all instances in the test set [66].

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (6.1.3)$$

R-Squared, also known as the coefficient of determination (or the coefficient of multiple determination for multiple regression), is a statistical measure that represents the proportion of the variance for a dependent variable that is explained by an independent variable or variables in a regression model [14]. In other words, the R-squared measures the strength of the relationship between your model and the dependent variables.

$$R - squared = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (6.1.4)$$

Finally, to better understand the trends and value distribution of the model's prediction, test set and training set values, the percentage histograms were developed and presented.

7. Performance Measurement Results

Table 7.1.1 shows various performance measurements throughout different years after publication for the prominent design. Figures 7.1.1 to 7.1.4 elaborates the citation frequency histograms for different years. The results of the baseline models can also be found in tables 7.1.2 to 7.1.4 and figures 7.1.5 to 7.1.13.

Table 7.1.1. Error values and measurements for Random Forest model

| | 1 year | 2 years | 3 years | 4 years | 5 years | 7 years | 10 years |
|----------|----------|---------|----------|----------|----------|----------|----------|
| MAE | 1.168999 | 1.89990 | 2.66408 | 3.43690 | 4.19492 | 5.62216 | 6.94813 |
| MSE | 43.389 | 71.769 | 124.990 | 216.971 | 356.035 | 786.339 | 1421.488 |
| RMSE | 6.58700 | 8.47164 | 11.17987 | 14.72992 | 18.86888 | 28.04174 | 37.70262 |
| R Square | 0.16119 | 0.22493 | 0.27972 | 0.303861 | 0.30706 | 0.29510 | 0.22525 |

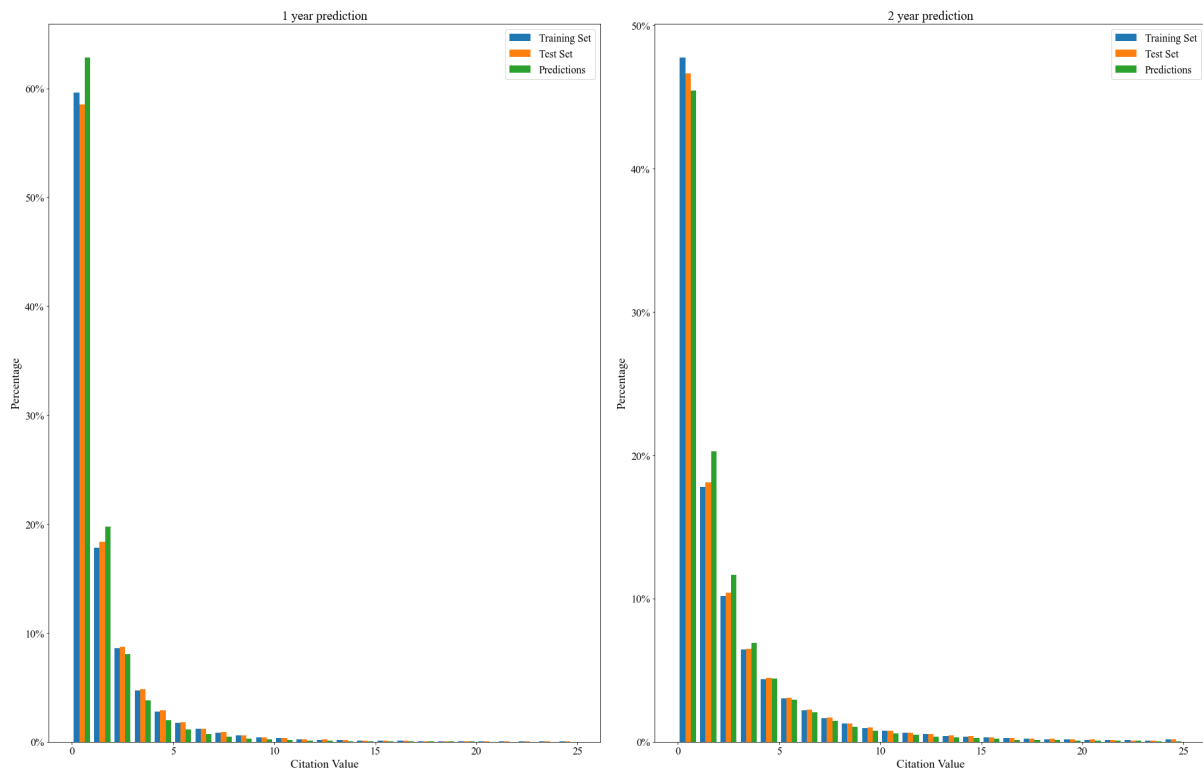


Figure 7.1.1. Percentage histogram for Random Forest model (1 year and 2 years prediction)

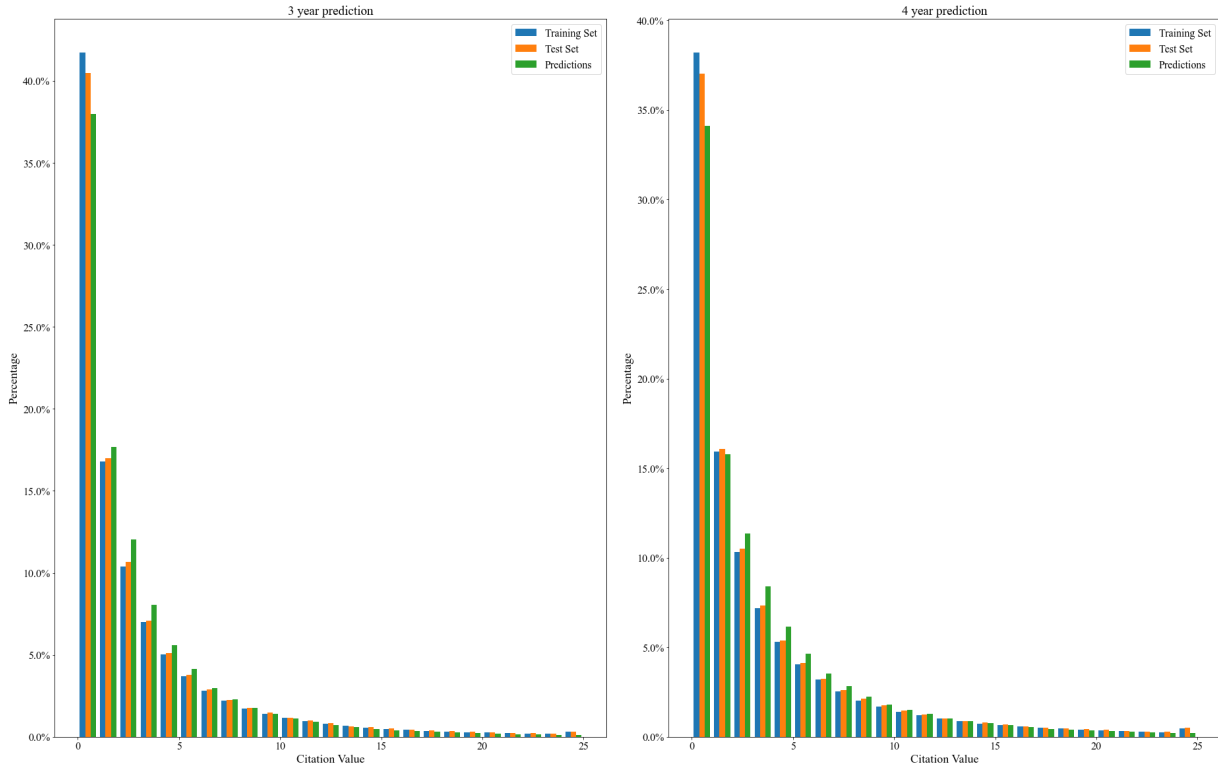


Figure 7.1.2. Percentage histogram for Random Forest model (3 and 4 years prediction)

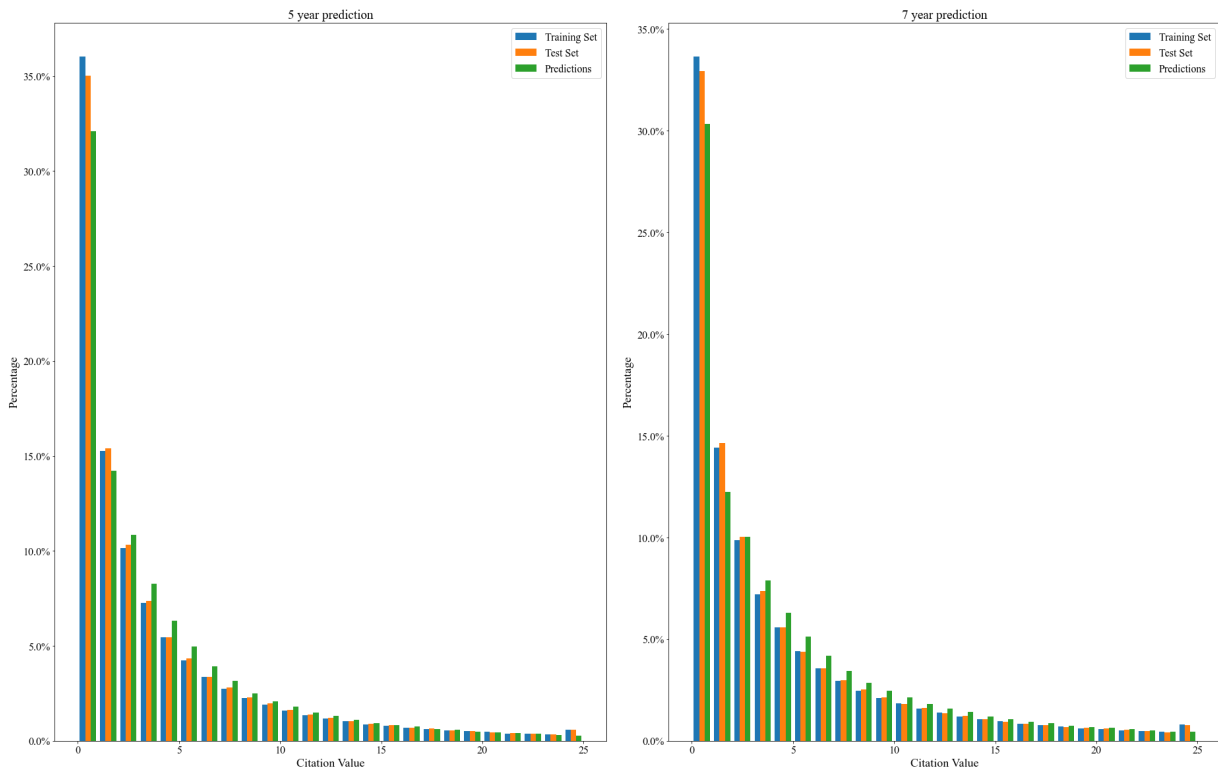


Figure 7.1.3. Percentage histogram for Random Forest model (5 and 7 years prediction)

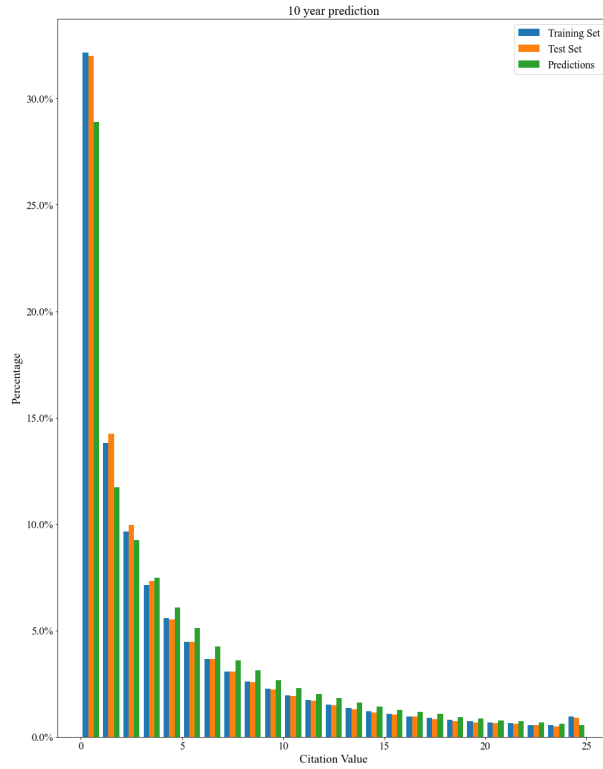


Figure 7.1.4. Percentage histogram for Random Forest model (10 years prediction)

Baseline models

Linear Regression

Table 7.1.2. Error values and measurements for Linear Regression baseline model

| | 1 year | 2 years | 3 years | 4 years | 5 years | 7 years | 10 years |
|----------|---------|---------|----------|----------|----------|----------|----------|
| MAE | 1.37992 | 2.42077 | 3.57197 | 4.78674 | 6.01039 | 8.37167 | 10.80188 |
| MSE | 49.432 | 84.025 | 152.809 | 271.789 | 447.725 | 979.027 | 1620.646 |
| RMSE | 7.03075 | 9.16653 | 12.36160 | 16.48604 | 21.15951 | 31.28940 | 40.25724 |
| R Square | 0.04436 | 0.09257 | 0.11941 | 0.12798 | 0.12860 | 0.12237 | 0.11670 |

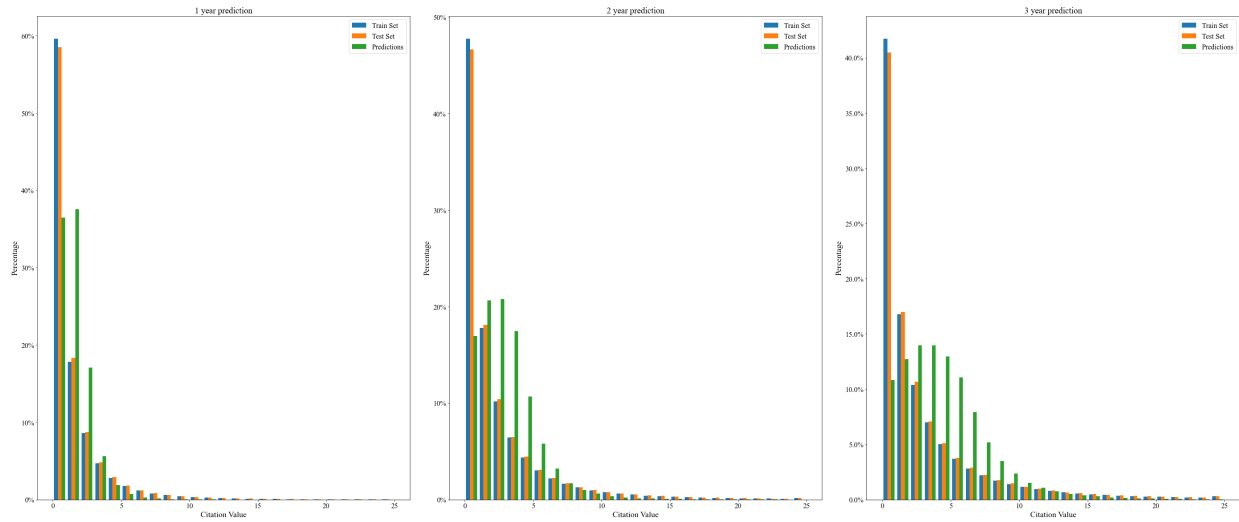


Figure 7.1.5. Percentage histogram for LR baseline model (1 year, 2 and 3 years prediction)

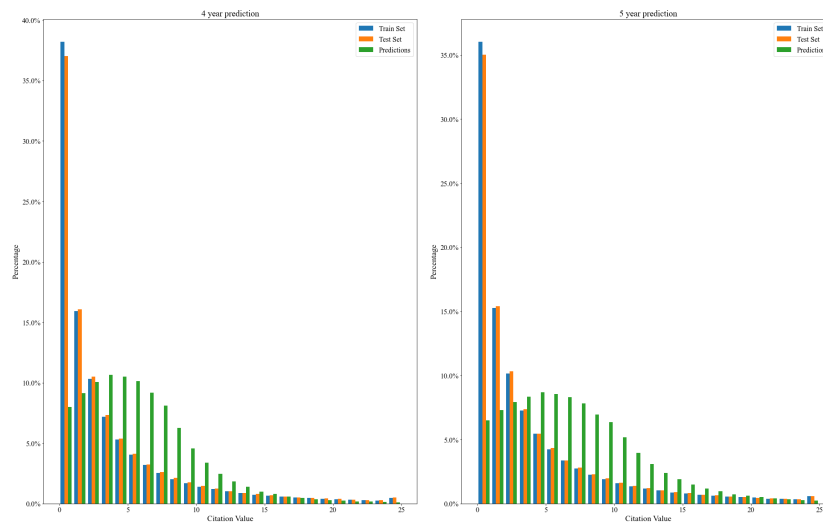


Figure 7.1.6. Percentage histogram for LR baseline model (4 and 5 years prediction)

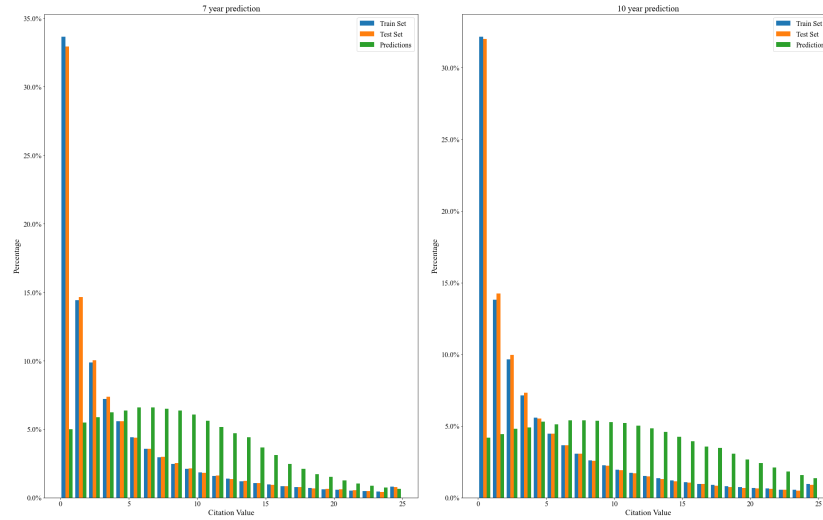


Figure 7.1.7. Percentage histogram for LR baseline model (7 and 10 years prediction)

SVM

Table 7.1.3. Error values and measurements for Support Vector Machine baseline model

| | 1 year | 2 years | 3 years | 4 years | 5 years | 7 years | 10 years |
|----------|---------|---------|----------|----------|----------|----------|----------|
| MAE | 1.22925 | 2.12890 | 3.10495 | 4.10677 | 5.07741 | 6.86114 | 8.18215 |
| MSE | 51.279 | 89.852 | 166.879 | 299.762 | 495.509 | 1083.034 | 1780.919 |
| RMSE | 7.16094 | 9.47904 | 12.91817 | 17.31364 | 22.26003 | 32.90948 | 42.20093 |
| R Square | 0.00865 | 0.02964 | 0.03833 | 0.03823 | 0.03560 | 0.02913 | 0.02936 |

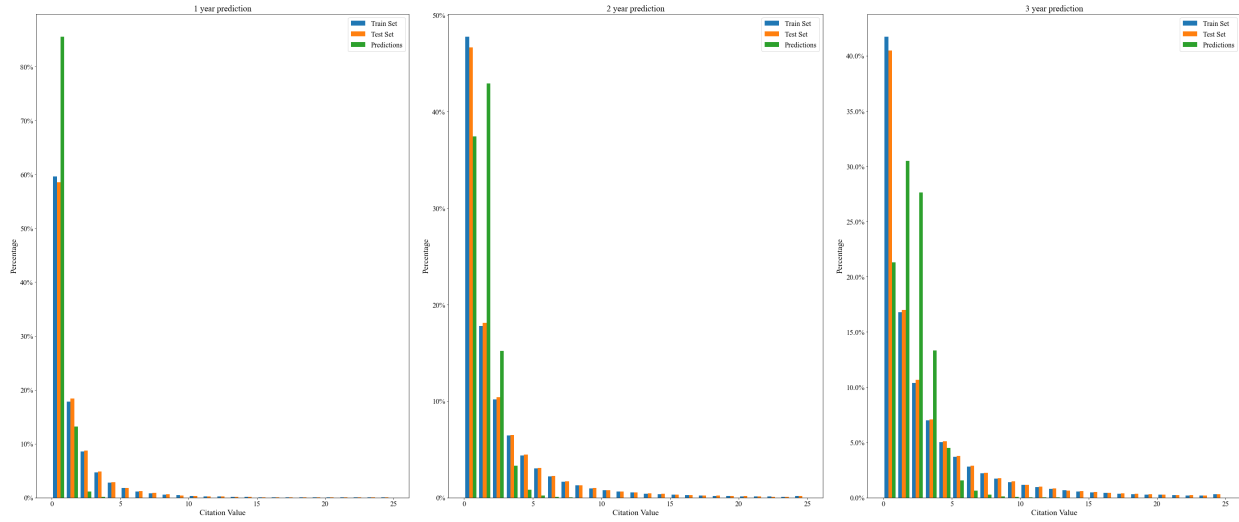


Figure 7.1.8. Percentage histogram for SVM baseline model (1 year, 2 and 3 years prediction)

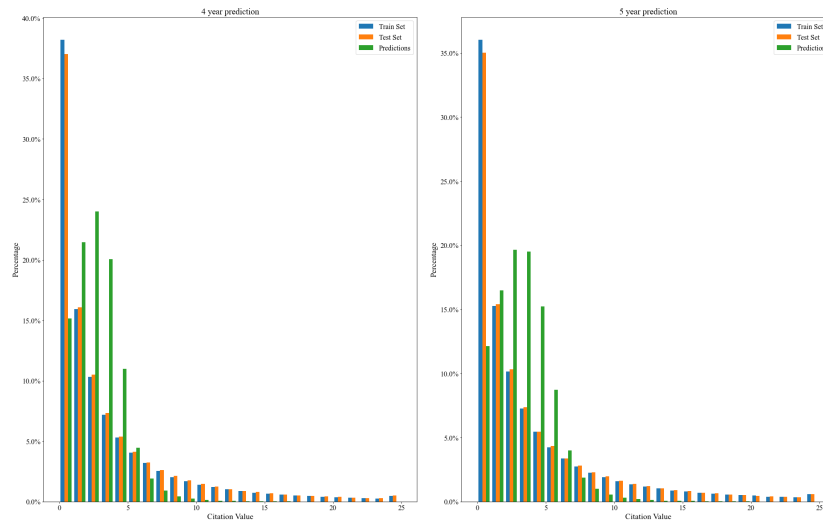


Figure 7.1.9. Percentage histogram for SVM baseline model (4 and 5 years prediction)

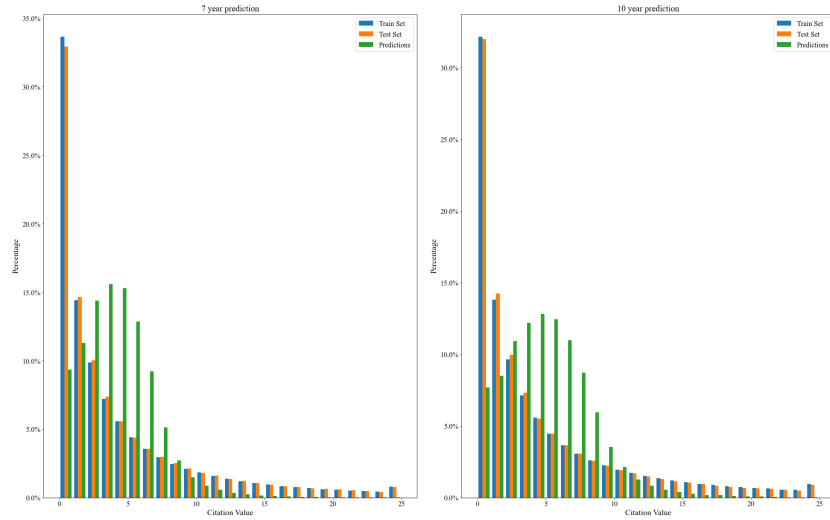


Figure 7.1.10. Percentage histogram for SVM baseline model (7 and 10 years prediction)

K-Means

Table 7.1.4. Error values and measurements for K-Means baseline model

| | 1 year | 2 years | 3 years | 4 years | 5 years | 7 years | 10 years |
|----------|---------|---------|----------|----------|----------|----------|----------|
| MAE | 1.25500 | 2.08803 | 2.96178 | 3.85689 | 4.73483 | 6.38549 | 7.90878 |
| MSE | 50.103 | 82.438 | 143.752 | 248.954 | 404.363 | 884.871 | 1547.107 |
| RMSE | 7.07833 | 9.07956 | 11.98965 | 15.77827 | 20.10877 | 29.74678 | 39.33328 |
| R Square | 0.03139 | 0.10970 | 0.17160 | 0.20124 | 0.21299 | 0.20677 | 0.15679 |

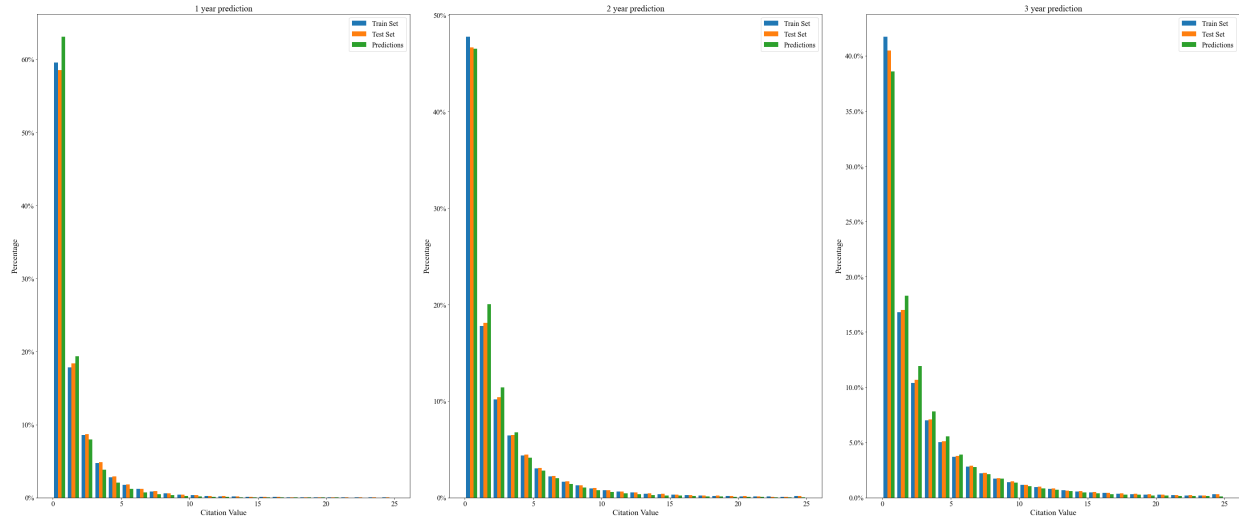


Figure 7.1.11. Percentage histogram for K-Means baseline model (1 year, 2 and 3 years prediction)

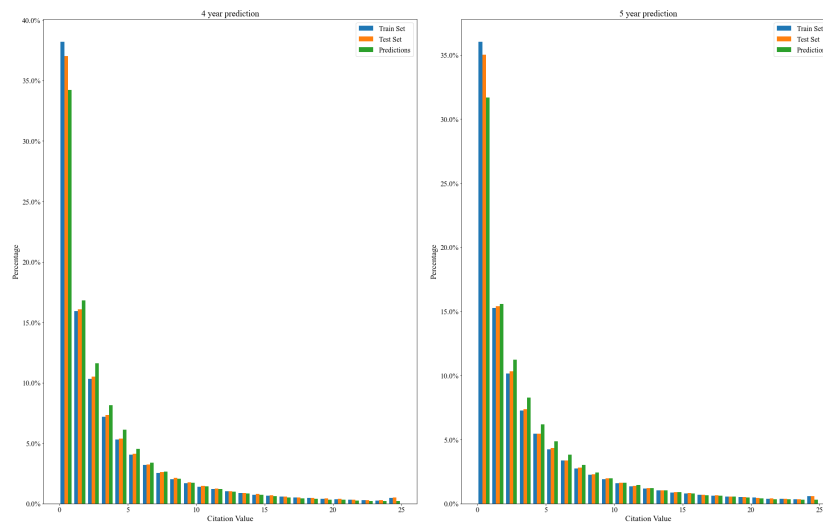


Figure 7.1.12. Percentage histogram for K-Means baseline model (4 and 5 years prediction)

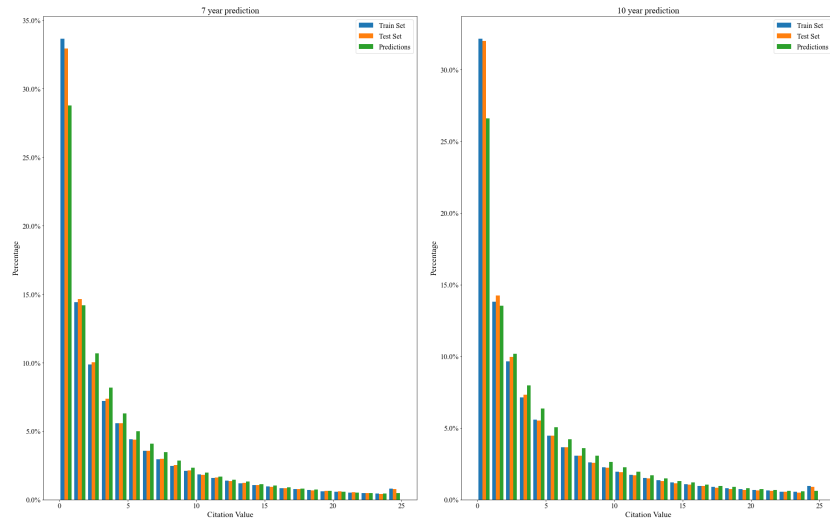


Figure 7.1.13. Percentage histogram for K-Means baseline model (7 and 10 years prediction)

8. Analysis of Performance

As mentioned before, we calculated the MSE, RMSE, MAE and R-Squared error values for the developed model. When looking at these values (Table 7.1.1), an increase in MAE is observed, which indicates an increase in the average value of residuals.

Similarly, an increase in RMSE values is detected indicating an increase in the square-root of the variance of residuals. Thus, by the increase of both MAE and RMSE, the model is less accurate with increase of year of citation. Another observation that can be taken from RMSE and MAE is the variance of the residuals. With the RMSE value being four to five times larger than the MAE value, we can see that the errors are not uniform throughout the prediction values and have a high variance. This indicates that on average the model performs well, and the error is smaller than the MAE. However, in some cases this error is noticeably higher.

R-Squared was another error value calculated for the model. which is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. The R-Squared value indicates how much the model explains the variability of the response data around its mean and it represents the proportion of the variance for a dependent variable that is explained by an independent variable. For the Random Forest model, it is seen that the highest value is at five-year citation followed by the year four; and the lowest is at year 1.

When looking at the percentage histograms of the yearly citations (Figures 7.1.1 to 7.1.4) we can see that the value distribution of the train set, test set and the prediction values by the model are fairly similar to one another. The distribution of the training and the test sets are the closest to one another, yet the predictions are following a similar pattern. It can be seen that in the earlier years, the smaller values have a higher than expected percentage value but as the number of years increases, this would shift to the higher values.

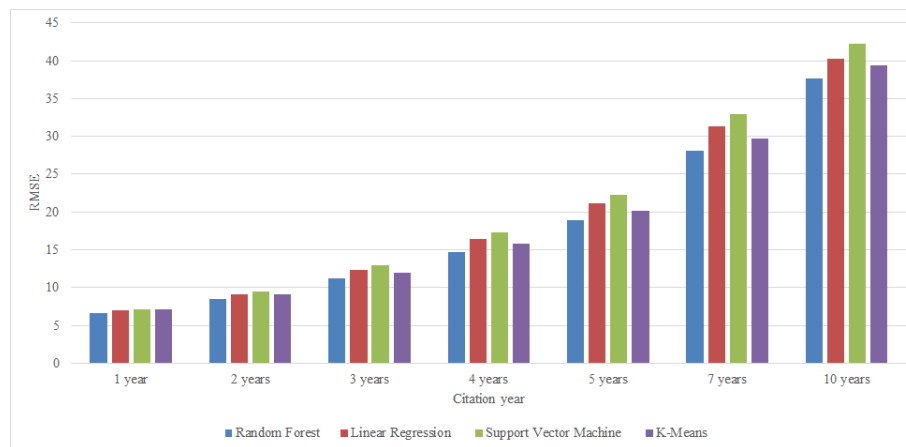


Figure 8.1.1. RMSE values of the models

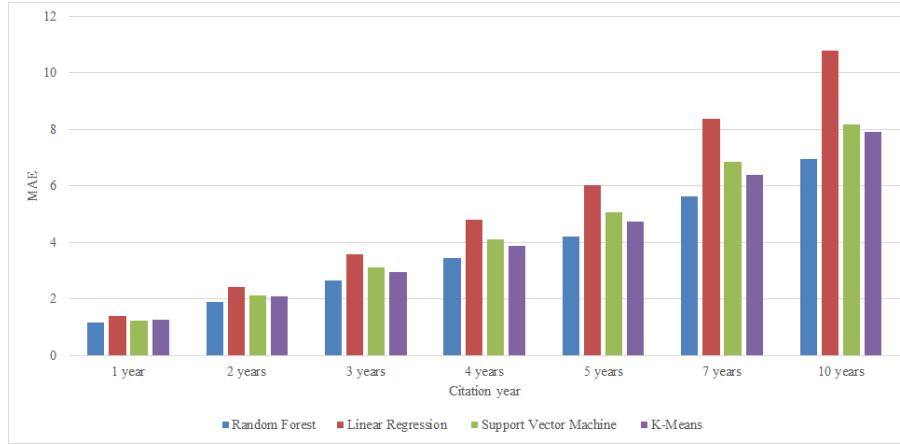


Figure 8.1.2. MAE values of the models

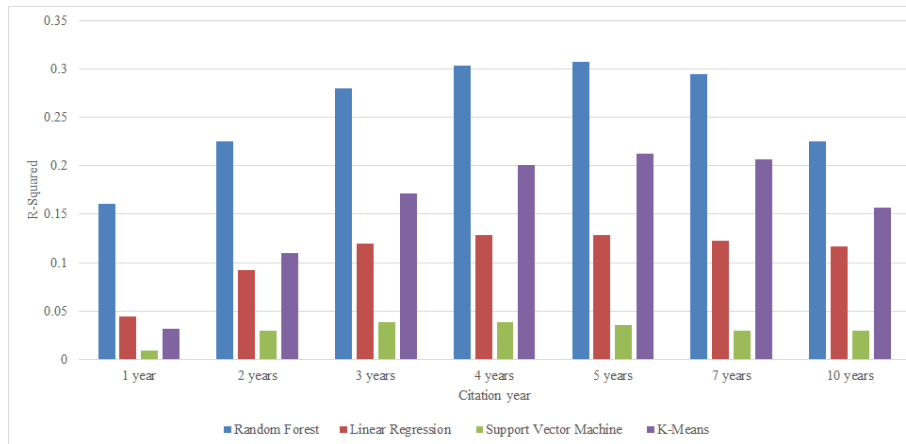


Figure 8.1.3. R-Squared values of the models

Analyzing the results from the baseline models reveals that the K-Means model is the best performing among the three baseline models. Both MAE and RMSE values are less than those of the other two models, and the R-Squared values are the highest. Moreover, when looking at the histograms of the three models (Figures 7.1.5 to 7.1.13), it can be observed that the K-Means model follows a similar distribution and trend to the test and training sets than the other models. Between the LR and the SVM models, it can be seen that LR has better values for RMSE and R-Squared whereas SVM performs significantly better in terms of the MAE. With the RMSE and MAE values of the LR model closer to one another, it can be noted that the variance of the error values is significantly smaller in LR model compared to SVM model. Based on the histograms in Figures 8.1.1 to 8.1.3, it can be concluded that the Random Forest model performs significantly better compared to the baseline models. Furthermore, it can be seen that the K-Means model has similar trends of values and is the second-best performing model. This implies that the nonlinear regression models perform significantly better than the linear regression models. This was also shown in Yan et al., 2012 and Yan et al., 2011 where multiple linear and non linear models were compared to one another.

9. Conclusions

In conclusion the current study successfully addresses the citation prediction problem. After analyzing multiple state of the art solution for this problem set, the research team decided to take an iterative solution approach. The dataset used during this study is the 12th version of Aminer citation database. This dataset was presented in the JSON format. During the retrieval process, the observations were fetched and organized based on the year. Data cleaning involved dropping the missing data, analyzing the outlier's nature and creating the initial dataframes. During the feature extraction step, the citations were re-evaluated as the number of the papers which reference the target paper. This enabled the algorithm to calculate the one to ten years impact of papers after their publication. During the model training session, the three candidate models were chosen as Random Forest decision tree, Gradient Boosting, and CART model. Furthermore, the Random Forest decision tree showed more promising performance results during the evaluation stage. The model achieved an MAE of 1.1689, MSE of 43.3886, and RMSE of 6.5870 on the 1 year citation prediction. After analyzing the histogram graphs, it was evident that both Gradient Boosting and CART models showed signs of higher bias towards some target values. The graphs for the Random Forest decision tree were consistent with the testing data and successfully tracked the data trends. Finally, feature engineering and hyperparameter tuning were performed to optimize the performance.

This research, like other studies, carries a set of limitations due to assumptions and the applied theories. The initial project objective was to predict the citation performance of a paper in the span of K years from publication. Based on our implementation, it was decided that the K range would be between 1 to 10 years. This is due to the idea that research papers can lose their academic relevance and significance over time. As technology and science advances, the older approaches become less applicable. Therefore, the older research papers receive less citations. Moreover, we assumed that the citations should happen using the references inside of the dataset. Therefore, there might be some discrepancies since the dataset may not cover all of the academic papers published during that period of time.

With the limited time and the large scope of study, difficulties and complications occurred. Some of these approaches required significant time and computational resources to be implemented. Pagerank algorithm and graph theory features are from such ideas which did not get implemented in the current model, however; should be included in the future work. Moreover, the scope of the project required analysis of the yearly citations while our dataset did not cover such information. As a result some assumptions had to be applied for prediction purposes. The future work would include study of other nonlinear models and features to discover the social dynamics which entice an author to reference another author.

References

- [1] O. Troisi, G. Maione, M. Grimaldi, and F. Loia, "Growth hacking: Insights on data-driven decision-making from three firms," *Industrial Marketing Management*, vol. 90, pp. 538–557, 2020.
- [2] R. Yan, C. Huang, J. Tang, Y. Zhang, and X. Li, "To better stand on the shoulder of giants," *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries - JCDL '12*, 2012.
- [3] K. P. Murphy, "Gaussian processes," in *Machine learning: a probabilistic perspective*, New York: The MIT Press, 2012.
- [4] A. C. Müller and S. Guido, "Neural Networks (Deep Learning)," in *Introduction to machine learning with Python: a guide for data scientists*, Sebastopol: O'Reilly Media, 2018.
- [5] X. Ruan, Y. Zhu, J. Li, and Y. Cheng, "Predicting the citation counts of individual papers via a BP neural network," *Journal of Informetrics*, vol. 14, no. 3, p. 101039, 2020.
- [6] R. Yan, J. Tang, X. Liu, D. Shan, and X. Li, "Citation count prediction," *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, 2011.
- [7] M. Schott, "Random Forest Algorithm for Machine Learning," Medium, 27-Feb-2020. [Online]. Available: <https://medium.com/capital-one-tech/random-forest-algorithm-for-machine-learning-c4b2c8cc9feb>. [Accessed: 10-Apr-2021].
- [8] T. Yiu, "Understanding Random Forest," Medium, 14-Aug-2019. [Online]. Available: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>. [Accessed: 03-Apr-2021].
- [9] AMiner|social@aminer.org, AMiner. [Online]. Available: <https://www.aminer.cn/citation>. [Accessed: 16-Mar-2021].
- [10] B. Boehmke and B. C. Greenwell, "Gradient Boosting," in *Hands-on machine learning with R*, Boca Raton: CRC Press is an imprint of the Taylor & Francis Group, an informa business, 2020.
- [11] B. Boehmke and B. C. Greenwell, "Decision Trees," in *Hands-on machine learning with R*, Boca Raton: CRC Press is an imprint of the Taylor & Francis Group, an informa business, 2020.
- [12] C. Sammut and G. I. Webb, in *Encyclopedia of machine learning*, New York, NY: Springer, 2011.
- [13] "RMSE: Root Mean Square Error ,," *Statistics How To*, 26-Jan-2021. [Online]. Available: <https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/>. [Accessed: 10-Apr-2021].
- [14] "Coefficient of Determination (R Squared): Definition, Calculation," *Statistics How To*, 28-Dec-2020. [Online]. Available: <https://www.statisticshowto.com/probability-and-statistics/coefficient-of-determination-r-squared/>. [Accessed: 10-Apr-2021].

Appendices

Appendix A.1

Results for CART Model

Table A.1.1. Error values and measurements for CART model

| | 1 year | 2 years | 3 years | 4 years | 5 years | 7 years | 10 years |
|----------|---------|---------|----------|----------|----------|----------|----------|
| MAE | 1.23085 | 2.12415 | 3.06031 | 3.95544 | 4.81040 | 6.40088 | 7.47600 |
| MSE | 50.732 | 87.116 | 159.827 | 283.935 | 464.750 | 1018.912 | 1705.767 |
| RMSE | 7.12266 | 9.33361 | 12.64229 | 16.85036 | 21.55807 | 31.92040 | 41.30093 |
| R Square | 0.01922 | 0.05919 | 0.07896 | 0.08901 | 0.09547 | 0.08661 | 0.07031 |

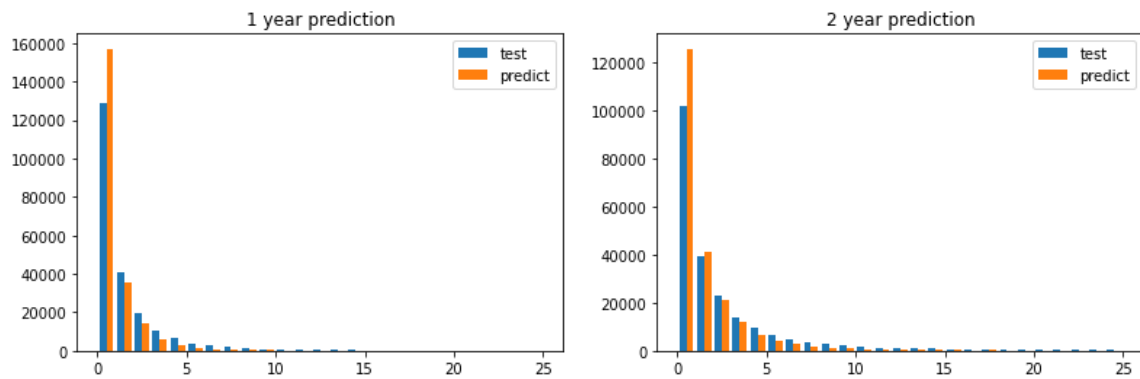


Figure A.1.1. Percentage histogram for CART model (1 and 2 years prediction)

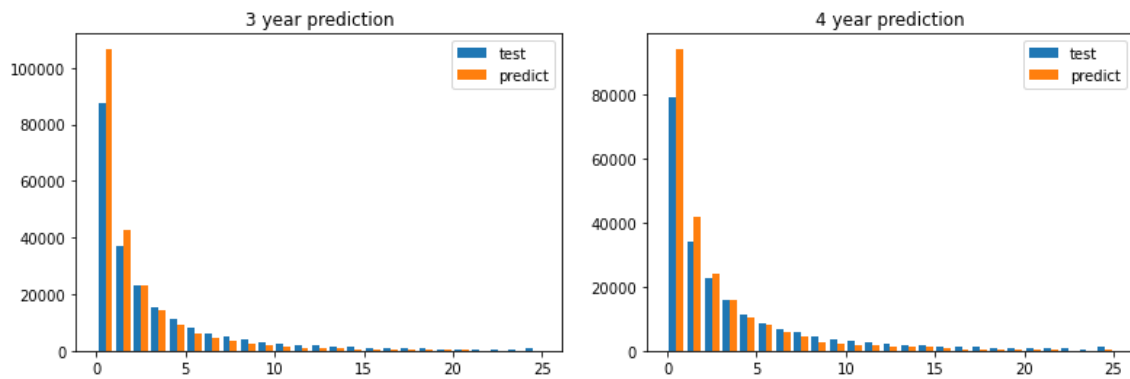


Figure A.1.2. Percentage histogram for CART model (3 and 4 years prediction)

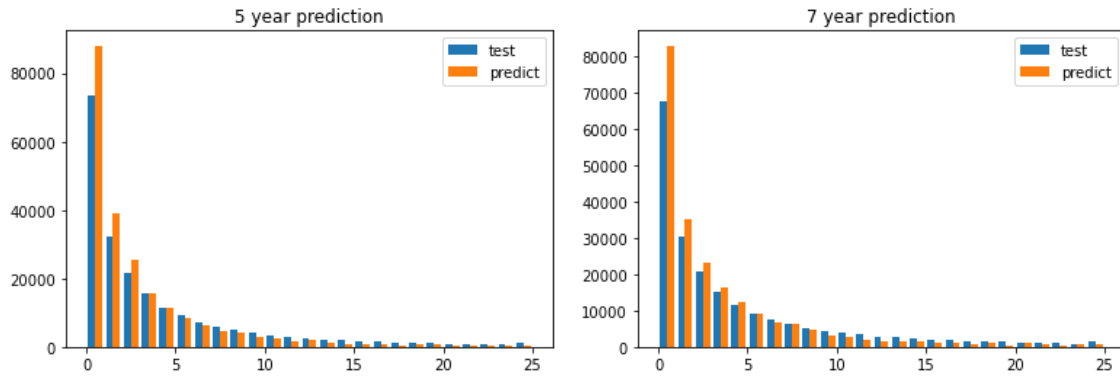


Figure A.1.3. Percentage histogram for CART model (5 and 7 years prediction)

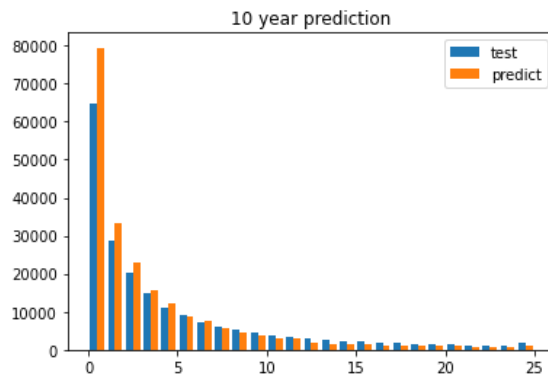


Figure A.1.4. Percentage histogram for CART model (10 years prediction)

Appendix A.2

Results for Gradient Boosting Model

Table A.2.1. Error values and measurements for Gradient Boosting model

| | 1 year | 2 years | 3 years | 4 years | 5 years | 7 years | 10 years |
|----------|---------|---------|----------|----------|----------|----------|----------|
| MAE | 1.17621 | 1.92152 | 2.73049 | 3.55472 | 4.30386 | 5.90718 | 7.199585 |
| MSE | 47.266 | 75.461 | 128.696 | 222.935 | 359.013 | 779.501 | 1361.988 |
| RMSE | 6.87500 | 8.68686 | 11.34442 | 14.93102 | 18.94764 | 27.91954 | 36.90513 |
| R Square | 0.08624 | 0.18505 | 0.25837 | 0.28472 | 0.30126 | 0.30123 | 0.25768 |

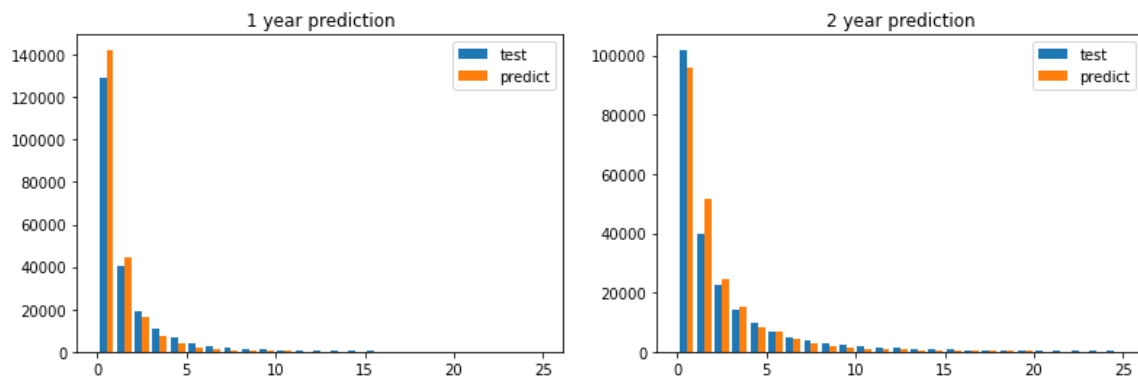


Figure A.2.1. Percentage histogram for Gradient Boosting model (1 and 2 years prediction)

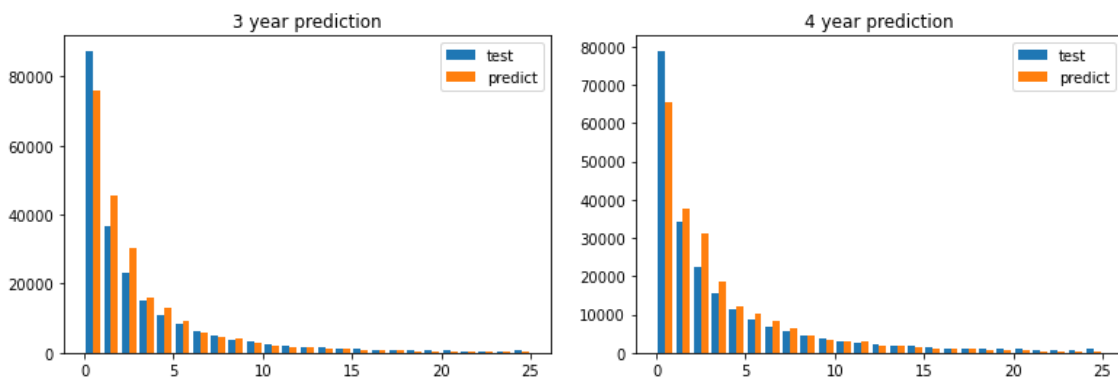


Figure A.2.2. Percentage histogram for Gradient Boosting model (3 and 4 years prediction)

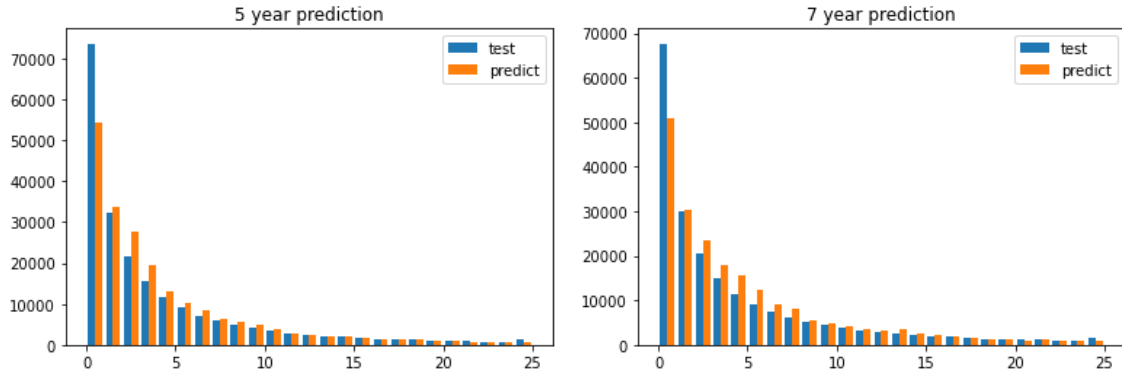


Figure A.2.3. Percentage histogram for Gradient Boosting model (5 and 7 years prediction)

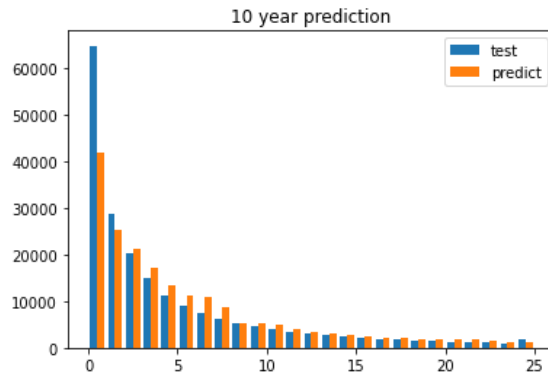


Figure A.2.4. Percentage histogram for Gradient Boosting model (10 years prediction)