University of Dublin

TRINITY COLLEGE

Time-Series Forecasting of Academic Careers

Dylan O'Driscoll

MCS Computer Science

University of Dublin, Trinity College

Supervisor: Prof. Dr. Joeran Beel

Submitted to the University of Dublin, Trinity College, Month, Year

School of Computer Science and Statistics

O'Reilly Institute, Trinity College, Dublin 2, Ireland

Declaration:

I, Dylan O'Driscoll, declare that the following dissertation, except where otherwise stated, is entirely my own work; that it has not previously been submitted as an exercise for a degree, either in Trinity College Dublin, or in any other University; and that the library may lend or copy it or any part thereof on request.

Signature:

Date:

Summary

There are many existing websites that allow access to databases containing a wealth of information on the past performance of academic researchers, including their year by year citation counts, publication counts, h-index and many more metrics. However, it can be difficult if not impossible for a human being evaluating all of these statistics to predict how a researcher will perform in the future. This can make it difficult for universities to decide which researchers to hire for certain positions, or for researchers themselves to know where they stand in relation to their colleagues. This dissertation investigates different ways of predicting how the evolution of different aspects of an academic researcher's career can evolve over time, using machine learning and time series techniques. Using data gathered from a public repository, different algorithms and strategies were compared in order to predict the citation count, publication count and future universities of academic researchers. For predicting metrics, 3 different time periods were analysed using two different prediction strategies. The three time periods were predicting their total citation count 10 years after first publishing based on data from their first 5 years, predicting citation count at 15 years based on the first 10 and at 20 years based on the first 15. In one strategy a single prediction was used to attain these values, and in the other the values of citation count, publication count and h-index for each individual year were predicted one year at a time, with predicted values of previous years used to predict the values for the next year. Future universities a researcher may go on to work at are predicted based on the order of the universities they have worked at previously

The results of these investigations yield a number of insights into an academic career. The h-index and publication count were found to greatly increase the ability of an algorithm to predict future citation count compared to citation count alone. The compound predictions were found to be less accurate than single predictions as the errors compound each year. Despite this, the compound h-index predictions were still reliable, demonstrating the strong predictive power of the h-index. Training algorithms on data from earlier years of a researchers career was found to decrease the performance of algorithms, indicating that they have little to no bearing on a researchers future performance. Finally, the order that a researcher attended universities in the past was found to have a huge impact on where they may go on to work in the future

Acknowledgements

To my mother and father, Margaret and Gary O'Driscoll, who have prioritised my education countless times since I was 3 years old, and without whom I could never have achieved as much as I have in all aspects of my life

To my brother and sister Conor and Megan O'Driscoll, whose company and sharp wit made the long days of running simulations and writing this dissertation bearable

To my girlfriend Ciara Gorman, whose own dedication to her academic pursuits and constant support inspired me to push on in the most difficult moments

To Conor Murphy, Conor Maguire, Yasir Mohamed and Darragh McCaffrey, whose friendship and camaraderie over the last 5 years has made my university days the most memorable of my life

Finally to my supervisor Joeran Beel, who was never short on ideas to constantly improve the quality of this work and was always ready and willing to share these ideas

Table of Contents

1 INTRODUCTION 1

1.1 Background 1

1.2 Research Problem 1

1.3 Research Goal 3

2 LITERATURE REVIEW 3

2.1 Motivations 3

2.2 Metric Prediction 4

2.3 Employment Prediction 6

2.4 Time Series Analysis 6

3 METHODOLOGY 7

3.1 Creating the Initial Model 7

3.2 Gathering the Data 8

3.3 The Dataset 9

3.4 Making Predictions 10

4 MACHINE LEARNING AND TIME SERIES BACKGROUND 14

4.1 Linear Regression 14

4.2 Random Forest 15

4.3 K-Nearest Neighbours 15

4.4 RMSE and MAPE 19

4.5 K-Folds Cross Validation 19

4.6 ARIMA Models 16

4.7 Multi-Layer Perceptron 18

4.8 LSTM 18

5 RESULTS AND DISCUSSION 20

5.1 Dataset Analysis 21

5.2 Single Metric Prediction 26

5.3 Compound Metric Predictions 34

5.4 Employment Prediction 46

6 LIMITATIONS AND FUTURE WORK 49

7 CONCLUSIONS 50

8 APPENDIX I 53

9 REFERENCES 56

1 Introduction

1.1 Background

A researchers' performance is something that needs to be assessed in many situations. When researchers apply for new positions, research grants or are in consideration for an award, several factors are considered. These include total number of publications and citations, the reputation of the journals and conferences in which these publications appeared, and some newer metrics such as h-index and impact factor [1]. Other metrics that are considered but not always formally analysed include previous grants awarded, PhD students supervised, previous roles and institutions worked at previously.

1.2 Research Problem

These metrics are all focused on the past performance of the researcher in question. They can only show how well a researcher has performed up to now, and it is difficult for a human analysing them to reliably predict how a researcher's career will evolve. There is some evidence to suggest that the metrics that are predominantly used are ill equipped to enable a human analysing them to make this kind of prediction. [1]

Figure 1, Figure 2 and Figure 3 show the citation count, h-index and publication count of Researchers A and B over the course of their careers. Researcher A's h-index has a clear positive trend, as does their citation count until the last 2 years. This would seem to indicate a healthy future career for them, albeit with a slight risk that their performance may decline. However, the future of the researcher's publication count is a lot less clear, with large increases and sharp drop-offs. This inconsistency in publication count may affect this researcher's future citation count and h-index in ways that are difficult to interpret from these graphs alone.

Researcher B has a much slower start to their career than Researcher A, but from 7 years after their first publication their productivity increased massively. Both are at the same point in their careers, with both having first published 14 years ago. This means that they may be applying for similar positions, such as assistant or associate professor. A person in charge of choosing which of the 2 researchers to hire would find it difficult if not impossible to decide whether to hire Researcher A or Researcher B using only these metrics. Even if such a decision could be made based on these metrics alone, there are even more factors to consider such as the reputation of the universities and journals that each researcher has published with, which further complicates the decision. In order to adequately evaluate these metrics, a more complex system or method is required than simply a human inspecting these metrics and making instinctual predictions.

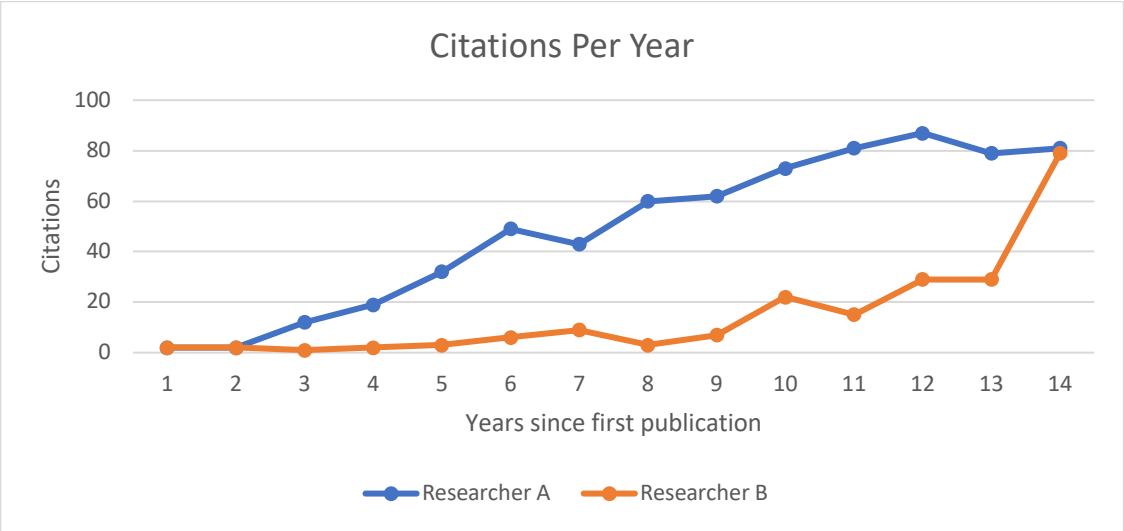

Figure 1: Citations per year for Researchers A and B

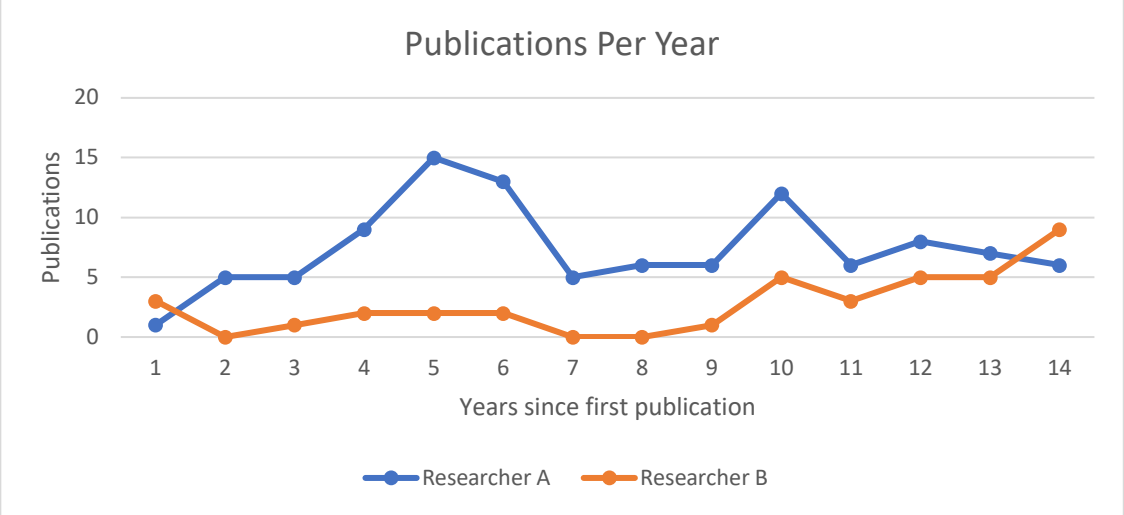

Figure 2: h-index each year for Researchers A and B

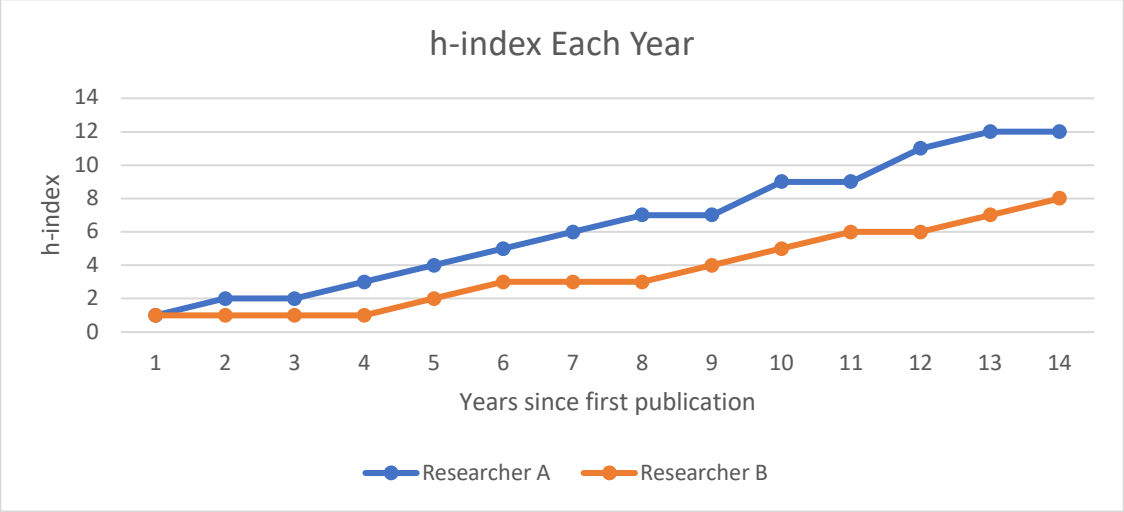

Figure 3: Documents per year for Researchers A and B

1.3 Research Goal

The research goal for this project is to investigate the ability of machine learning and time series techniques to predict the evolution of academic careers, with the ultimate aim of creating a method that can reliably predict how certain metrics of an academic career will change as time passes. For example, a researcher may input their citation count, document count and h-index for each previous year of their publishing career, as well as the previous universities they have worked at. The method would output a prediction for each of these metrics 5 years into the future as well as a year by year breakdown of them. It would also output a list of universities that the researcher is likely to go on to work at. Multiple machine learning algorithms and other types of predictions will be experimented with in order to achieve this goal.

One of the main reasons that machine learning techniques have become so popular in recent years is their ability to handle multiple different features and combine them into a cohesive model that produces predictions that would impractical to calculate by hand or on older computers. The multiple features described in Section 1.2 are difficult for a human to analyse but are very well suited to assessment by machine learning techniques.

The first step was to create a model for measuring academic performance that is as comprehensive as possible. Machine learning algorithms will then be used on a simplified version of this model, accounting for only previous work history and citation counts. This first iteration will then be expanded to include more features from the original model as possible to improve the standard of predictions.

2 Literature Review

2.1 Motivations

Attempting to predict future events is one of humanity's oldest traits. As early as the 7th century BC, Greek citizens would consult the Oracle of Delphi for prophecies about their own lives [2]. In modern times, people have begun to use machine learning and other statistical methods to attempt a large range of predictions, from diagnosing diseases [3] and predicting stock market trends [4] to more personal predictions like recommending articles to users based on their past reading history [5]. This has been facilitated by the constant increase in computing power available to the public, allowing large datasets to be analysed and predictions made in a reasonable amount of time using algorithms that would have taken days or weeks only a decade ago. As machine learning techniques increase in popularity, more and more industries are attempting to leverage these techniques to improve their decision making.

Academia is no exception to this trend of wanting to see into the future. van Dijk et al. [6] used linear models to predict how likely a researcher is to be successful in applying for role as Principal Investigator, using a dataset with over 500 metrics describing the careers of 25,000 medical scientists. Daud et al. [7] sought to predict from as early a point as possible whether or not a researcher is what they define as a 'rising star', using a number of different algorithms to see which yield the best results. More generally, machine learning has been used on a wide range of topics in academia. Instead of focusing on the researchers themselves, a study by Fu et al. [8] describes a model that can predict year by year citation counts for publications in the biomedical field up to 10 years after it is first published, using only data available on publication. The machine learning methods used make use of both content based and bibliometric features. It has also been used to extract the authors and affiliations from a given publication and establish relationships between different articles [9].

The motivations behind being able to reliably make these types of predictions are two-fold. If you are attempting to decide between multiple candidates for a job opening, being able to reliably predict how they will perform in the future is a huge help in making that decision. The same also applies to those attempting to get hired. Seeing which candidates have better predicted careers can help inform decisions about their own career and how to give themselves the best chance at success.

2.2 Metric Prediction

The most popular predictions made in relation to academia are regarding the metrics that describe a researcher's career. One of the most important metrics, both as a tool for predicting other metrics and as something to be predicted, is the h-index. Since it was first proposed by J.E. Hirsch in 2005 [10] it has been used by many others to predict how it and other metrics like citation count and publication count will change over the course of a researcher's career. According to [10], "A scientist has index h if h of his or her N_p papers have at least h citations each and the other $(N_p - h)$ papers have h citations each". Hirsch found that the h-index gives a more accurate view of a researcher's career than citation count alone, which can be artificially inflated by a researcher having a small number of successful papers or by having a minor role in co-authoring multiple papers.

The h-index has become a very popular metric since it was first introduced, with many studies attempting to quantify its ability to represent past performance and predict future performance. One of the most important studies in in this area was conducted by Hirsch himself [11]. In this paper, he assesses its use at predicting how other metrics will change over time. No machine learning techniques are used to carry out these assessments. Instead, the metrics of 50 physicists who first published between 1978 and 1982 are measured after the first 12 years and

first 24 years of their careers. Each combination of these metrics (12-year h-index vs 24-year citation count, 12-year publication count vs 24-year publication count etc.) was analysed and the correlation between them calculated. Hirsch showed that the h-index had the highest ability to predict its own future value of any of the metrics analysed and could predict other metrics more accurately than any other single metric. The h-index was also the subject of a study by Acuna et al.[12]. This study used a simple linear regression to predict a researcher's h-index for each year up to 10 years in the future, using the number of articles a researcher had written, their current h-index, the number of years since they first published, the number of distinct journals they had published in and their total number of publications as training features.

The h-index is also used in a paper by Bertsimas et al which seeks to predict publication count and citation count 9 and 16 years after a researcher's first publication [13]. This paper uses a random forest model based on citation count, h-index and both A-journal and total publication count after 5 years as a baseline model. Predictions from this model are then compared to predictions based on a network model based created by the authors of the paper. To create this network, the authors first processed 198,310 papers authored by 136,313 authors to create two networks. In the citation network, nodes represent papers and directed edges represent one paper citing another. In the co-authorship network, nodes represent authors and undirected edges represent authors working together on a paper. These networks were combined with undirected edges connecting authors to papers they authored. Undirected edges in the co-authorship network were replaced with a pair of directed edges to facilitate the merging of the 2 networks. This network was created for each year between 1975 and 2012 to illustrate how the researchers' careers evolved over that time. The authors created a number of metrics based on this network to describe the career of individual researchers. Combining these metrics with standard metrics such as citation count, h-index and publication count, the authors again used a random forest algorithm compare with their baseline.

Another study by Amin Mazloumian disputes the claims made by Hirsch [11] that the h-index has the most predictive power of all metrics. It proposes instead that citations per year, not included in the original study, is a more effective metric than h-index for predicting future citation count, and that using some other metrics in conjunction with these can actually decrease predictive power. Mazloumian also proposes the g-index, an alternative to the h-index, which is the highest value of g papers by a researcher that have received g^2 citations [14].

It is clear from these papers and others like them that citation count, publication count and h-index are very popular and usful metrics. This is partly because they are both simple to calculate and it is easy to understand what the values of these metrics mean for an academic

career. They are both important as values to predict and as data used to make these prediction But there are other features that have also been shown to have predictive power. A study of 182 biologists by Laurance et al. [15] found a strong correlation between pre-PhD publication count and overall productivity (defined here by publication frequency) as well as weaker correlations between productivity and gender, university prestige and whether or not English was a researcher's native language.

The availability of different metrics played a huge role in choosing which ones to include in this project, as it did in the studies described above.

2.3 Employment Prediction

Much of the efforts to predict future employment for researchers have focused on what positions or accolades a researcher is likely to receive, rather than the universities they may go on to work in. As mentioned above, van Dijk et al. [6] attempted to predict whether a researcher would be successful in applying for a Principal Investigator position based on the number of papers published by a researcher and the impact factor of the journals in which they were published. Bertsimas et al. [13] used their network model to predict which researchers would be selected for tenure positions. It's possible that the actual universities offering these positions may not be of interest to many academics, hence the lack of research in this area. Nonetheless, I believe predicting a future place of employment reliably to be a good demonstration of the power of machine learning techniques.

2.4 Time Series Analysis

As mentioned previously, machine learning techniques have been applied to a wide range of areas. Experiments carried out in one area may contain solutions to problems in others. Time series analysis has long been an area where people have sought accurate predictions. A time series is a list of measurements of the same value at regular intervals. These measurements could be anything from air pollution levels in a city to the amount of people queueing at a bank. A researcher's citation count, publication count and h-index each year can all be viewed as time series and so popular time series algorithms can be used to make predictions about them. One of the most popular time series prediction algorithms is the Autoregressive Integrated Moving Average (ARIMA) model [16, 17]. This model analyses just one time series at a time and so cannot be used on multivariate inputs, which may put it at a disadvantage to other algorithms that can make use of multivariate input.

There have been successful attempts at applying machine learning techniques, in particular Artificial Neural Networks, to multivariate time series problems as varied as the availability of

water resources [18] and flour prices [19]. These approaches could prove useful for predicting the future values of various metrics.

Neural networks are not the only machine learning algorithms that can be applied to time series. Sinn M et al. [20] used several versions of a K Nearest Neighbours style algorithm to predict when a bus will arrive at a certain stop based on its journey so far. The cumulative number of metres the bus had travelled each minute were used as features to calculate the Euclidean distance between different journeys on the same route. The times taken for buses to reach future stops on similar journeys were used to predict how long a bus mid-way through a journey would take to reach those stops. The team behind this paper went on to implement this technique with Dublin Bus.

Predicting academic success is rarely considered as a time series problem, so these methods may yield different results to the studies referenced in Sections 2.2 and 2.3.

3 Methodology

To achieve the stated research goal, a plan was created

1. Create model containing all factors that may contribute to researcher performance.
2. Develop a method of parsing the publicly available data from websites like Google Scholar, Microsoft Academic or Scopus, either through a pre-existing API or HTML parsing. In the end only Scopus was used for reasons detailed below.
3. Based on which features are most readily available and most commonly used in relevant literature, create a dataset using the methods created in step 2.
4. Create first iteration of the method which makes predictions based on single variables. These predictions will be used as baseline results.
5. Create second iteration which makes predictions based on multiple variables
6. Refine this iteration as much as possible, including as many relevant features as possible before the deadline.

3.1 Creating the Initial Model

After consulting the relevant literature, citation count, publication count and h-index were identified as being the most important metrics. Other metrics that could possibly affect career development were included as well. The model, as can be seen in Figure 4 is split into 2 sections, personal details and academia. The academia section is further split into research, publication and teaching. These divisions would have no bearing on a machine learning algorithm but make it easier for someone seeking to understand the selection of features what they mean and why they are important.

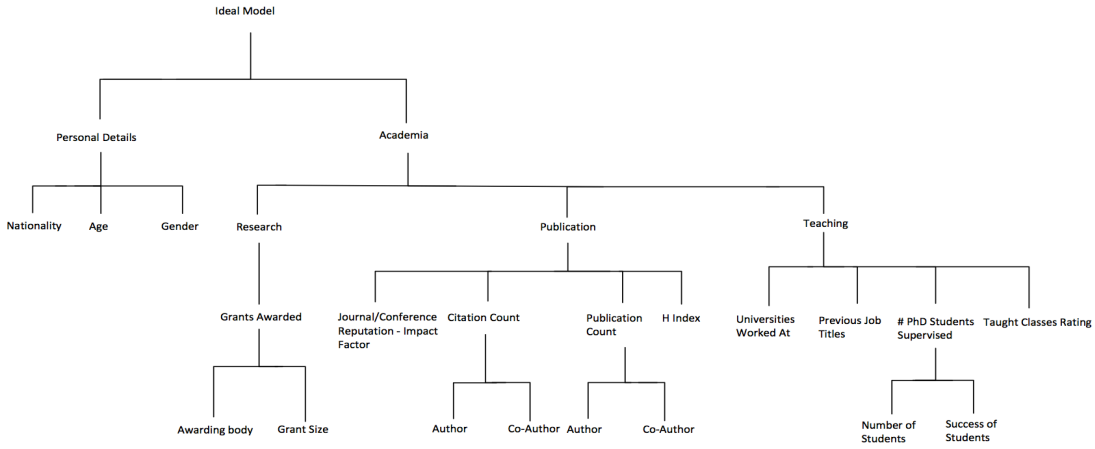

Figure 4: Complete Model of an Academic Career

3.2 Gathering the Data

Many of the metrics in the ideal model cannot be used to compare researchers in different fields. A h-index that might represent an impressive career for a biologist may represent an average career for a mathematician. The first step therefore was to create a list of researchers from the same discipline whose career histories could be used as training data. The discipline itself was not important so long as it was the same for all researchers, so computer science was chosen. The list was created by simply searching for 'Computer Science' in Harzing's *Publish or Perish* [21] tool, exporting the results to a CSV file and extracting the author names. This resulted in a list of 2788 Computer Science researchers whose career histories could be used to train and test machine learning algorithms.

After compiling this list, the next step was to create a dataset containing the career histories of these researchers. A list of online databases that could potentially be used was compiled from my own research and from suggestions by my supervisor. The databases, while free of charge and very useful for both this use case and for other people all over the world, rely on PDF parsing to collate their statistics. These parsers can be error prone and so the figures generated based on them may not be entirely accurate [22-25]. As well as this, some of the sources that they publication data from are publicly maintained and therefore can be prone to manipulation [26]. Despite these flaws, the ease of access to these online databases outweighed possible risks of inaccurate data.

The databases initially identified as being potentially suitable were Google Scholar, Research Gate, Mendeley, Orcid, Semantic Scholar, Microsoft Academic, Academia.edu and Scopus. Of these, only Scopus, Semantic Scholar, Mendeley and Orcid offered an API. If the other

websites were to be searched then a HTML parser would have to be developed on top of the rest of the project, which would consume a lot of time that could be used working on the machine learning aspect of the project. Of the four websites that offered an API, only Scopus maintains its own database. The other websites rely on researchers signing up and entering their own details and were more like social networks than pure databases. This means that Scopus has information on much more researchers, with the trade-off that in cases where a researcher had taken the time to fill out their details on another website, the other website would have a much more detailed history for that researcher. Indeed, when searching each website for the researchers on the list, 87% of them could be found on Scopus, with the next highest being 54% on Mendeley. There are always arguments to be made for quality vs quantity of data but in this scenario the difference in quantity was too steep, especially considering that the core metrics of citation count, document count and h-index were all available for every author on Scopus. Another factor to consider is that Scopus does not discriminate regarding which researchers it collects information on. For information to appear on other websites, a researcher would have to make the conscious decision to sign up to that website, which could be an indication of some other factor in their careers that could not be accounted for in the scope of this project. For these reasons, Scopus was chosen as the database from which the information for this project would be gathered.[1]

The list of names was then fed into 4 separate scripts, one to search for the employment history of each researcher and 3 to search for the year by year citation count, document count and h-index for each year between the publication of their first document and 2017 (The year 2018 was incomplete at the time of retrieval so any metrics for that year would not have been representative). These were the metrics most commonly available and the ones most often used in the papers discussed in Section 2, so they were deemed good enough for initial predictions at least.

3.3 The Dataset

The scripts for downloading citation counts, document count and h-index created three CSV files, one for each metric. Any duplicate entries (i.e. entries with identical features and similar or identical names) were removed. This is a very simple workaround for the name disambiguation problem that previous efforts at predicting these metrics have encountered [7, 13]. Each of these 3 files has the same basic structure. The first column contains the name of

[1] The data was downloaded from Scopus API between January 29th and April 18th, 2018 via http://api.elsevier.com and http://www.scopus.com.

the researcher, and the rest contain the citation count, document count or h-index for each year since the year that their first document was published. For example, this line in the citations file

Serena Villata,4,25,14,25,52,68,106,95,113,120

represents the researcher Serena Villata, who received 4 citations in the year that her first document was published, 25 the year after that, 14 the year after that etc.

The employment history dataset is structured in a similar way to the other datasets. Each line contains the name of a researcher and the universities that they have worked at, identified by their Scopus identification number. For example, the line

Andrew W Fitzgibbon,60027272,60026851,60022148

represents the researcher Andrew W Fitzgibbon, who worked at the University of Edinburgh, University of Oxford and University College London. The dataset contains 1738 different universities, attended by 1878 researchers The average number of universities worked at be researchers in the dataset is 4.1, with a wide standard deviation of 7.3. Both datasets are further analysed in Section 5.1.

3.4 Making Predictions

3.4.1 Metric Predictions

3.4.1.1 Baseline Predictions

The first step was to create a baseline of predictions based on a simple method that later predictions could be compared to. This baseline consisted of training various machine learning formulas to make 3 predictions: A researcher's total citation count 10 years since their first publication based on the citation counts of their first 5 years, their citation count after 15 years based on the first 10 years and their citation count after 20 years based on their first 15 years. These 3 predictions cover a large portion of a researcher's career and the results inform what the algorithm could be used for. If predictions are accurate at predicting 10 year citation count but not 15 year citation count then it would be useful for assessing the careers of prospective assistant professors but not someone applying for a full professor position. As these predictions were only intended to be used as a baseline, the algorithms were only trained on citation counts. Further predictions, described in Section 3.4.1.2, make use of multivariate features.

To facilitate these predictions, 3 separate files were created based on the citation dataset. One file had the first 5 years of each researcher's citation data followed by their total citation count after 10 years. The second has data for 10 years plus the 15 year total and the last has data for 15 years plus the 20 year total. This line in the file containing the first 5 years of citation data

Andrew W Fitzgibbon,0,0,1,1,24,652

represents Andrew W Fitzgibbon, who received 0 citations in his first and second years of publishing, 1 in his third and fourth, 24 in his fifth year and had received 652 citations in total after 10 years since first publishing.

Three algorithms were used to make these baseline predictions: Random Forest, K-Nearest Neighbours and Linear Regression. Each of these algorithms used the 10, 15 or 20 year total as their target variable and the other years specified in the dataset as the feature vector. The algorithms were implemented using the Scikit-learn[27] Python library, and their performance was measured by calculating the Root Mean Square Error (RMSE) and Mean Average Percentage Error (MAPE) of the predictions made. Due to the relatively small size of the datasets, K-Folds cross validation was used to help to ensure that these results aren't skewed heavily by outliers or by randomly selected test and training data that happen to provide better results. Two different result aggregating schemes are provided by Sci-kit Learn and both were compared. One returns the uniformly weighted average of the value of the target variable from each of the k-nearest neighbours, and another returns the weighted average of this variable based on the distance between the neighbour and the researcher being analysed. These machine learning techniques and concepts described in detail in Section 4.

3.4.1.2 Multivariate Predictions

The baseline predictions are fundamentally limited by only using citation data to train the algorithms. As is the case with many machine learning and time series problems, future values are often dependent on the previous values of multiple variable and not just their own. The multivariate predictions makes use of citation count, publication count and h-index to train the algorithms. Two sets of predictions are made in this second iteration. Predictions were made using the same three algorithms as were used for the baseline predictions as well as three others: An Autoregressive Integrated Moving Average (ARIMA) model, a standard neural network model based on a multi-layer perceptron and a recurrent neural network based on long short-term memory units (LSTMs). Both of these algorithms are described in further detail in Section 4.

The first set of predictions is the same as the baseline predictions from the previous section. Algorithms are trained on the first 5, 10 or 15 years of a researcher's career and make predictions for their total citation count 5 years later. The difference here is that the algorithms are now trained on the h-index, publication count and citation count for each year of a researcher's career and not just citation counts. To facilitate these predictions, 4 datasets similar to the ones used for the baseline predictions were created. Each line of each dataset represented a researchers h-index, document count and citation count for the first 5, 10 or 15 years since their first publication, with the target variable being either their total citations in

years 10, 15 or 20 or the difference between these values and the researcher's total citation count in years 5, 10 or 15. Predicting differences is common in time series analysis and the motivations behind it are explained below. Two versions of these datasets were created, containing the individual values for each metric per year and the other containing cumulative values, to see if different representations of the same data would perform differently. This line from the 5-year version of the individual values file

Dragos Niculescu,0,2,5,7,12,1,6,4,1,2,0,30,95,225,284,3075.

represents Dragos Niculescu whose h-index for the first 5 years of his career were 0, 2, 5, 7 and 12, who published 1, 6, 4, 1 and 2 publications and received 0, 30, 95, 225 and 284 citations and by his 10th year of publication he had received 3075 citations. The same researcher is represented in the differenced version of the file by the line

Dragos Niculescu,0,2,5,7,12,1,6,4,1,2,0,30,95,225,284,634,3075,2441

which shows the same data but also includes the researcher's total citation count after 5 and 10 years and the difference between them. The cumulative versions of these lines are

Dragos Niculescu,0,2,5,7,12,1,7,11,12,14,0,30,125,350,634,3075

and

Dragos Niculescu,0,2,5,7,12,1,7,11,12,14,0,30,125,350,634,634,3075,2441

respectively. The values for the h-index were kept as the values for individual years as the concept of a cumulative h-index does not translate in the same way as a cumulative publication or citation count. The h-index is a measure of a researcher's performance over their whole career, not just one year, so adding multiple years together provides no additional information.

The second set of predictions takes the same inputs as the other set for training but instead of predicting citation count 5 years in the future, it attempts to predict all three metrics one year into the future. These predicted values are then combined with the original training data to predict these values two years after the original cut-off point. The process is repeated to gather predictions for the first 5 years after the cut-off point. For all of these predictions, different numbers of previous years were used as input and the results were compared to find the optimal number for each algorithm.

3.4.2 Employment Prediction

3.4.2.1 Baseline Predictions

As with the metric predictions, a baseline is required for the attempts to predict future employment. To create this baseline, the employment history dataset was analysed with multiple variants of a k-nearest neighbours style algorithm. These baseline predictions do not take into account the order in which each researcher joined each institution, so each instance in the dataset is modified to create multiple versions of that instance, each with a different institution as the target. For example, if a researcher has worked at 5 institutions, 5 variants

would be created, each with a different institution as the target and the remaining four as the features for that instance. Only researchers with 5 or more previous employers were regarded for this analysis, as smaller numbers would lead to almost random guessing which would mask the true performance of the algorithm. As K-Nearest Neighbours trains on the whole dataset for each prediction, K-Folds cross validation was not used.

Usually the similarity between feature vectors for k-nearest neighbours is calculated using Euclidean distance. This was not appropriate for the employment dataset as although each institution is represented by a numeric ID, these numbers are not derived from any feature of the institution. An implementation was considered where instead of representing employment history as a set, it could be represented as a binary vector where each column indicates whether a researcher has worked at that institution or not. The Euclidean distance between these vectors would then be used to determine the k-nearest neighbours and their weights. This was not attempted as the resulting matrix would be too sparse and the distances would take too long to compute. The Jaccard index, a common way of examining the similarity between mathematical sets, is therefore used instead of Euclidean distance in order to calculate similarity. The Jaccard index of 2 sets A and B is defined as $|A \cap B| \div |A \cup B|$.

After the k-nearest neighbours are identified, a list of all the institutions those researchers had worked at is compiled and weights were assigned to each institution. Multiple ways of assigning these weights are implemented and compared. One variant simply assigns the weight as the number of the k nearest researchers in who had worked at that institution. In another, this simple weight was modified using Term Frequency-Inverse Document Frequency (TF-IDF) to increase the weights of rarer institutions and decrease the weight of more popular ones. TF-IDF was originally designed as a way of assigning documents to categories based on the frequency of terms in those documents. The formula for TF-IDF is $tf(d_i, t_j) \times \log(\frac{n}{|d_{t_j}|})$ where $tf(d_i, t_j)$ is the frequency of term j in document i, n is the number of total documents and $|d_{t_j}|$ is the number of documents that contain term j. When applied to this use case, the term frequency is the weight assigned to the institution and each authors employment history is a document. A third strategy was tested where the weight was the sum of the Jaccard indices between the researcher being analysed and the k-nearest neighbours, and a fourth applied TF-IDF to this weight. Any list which had the target institution as one of its top 10 weighted entries was considered a success.

3.4.2.2 Sophisticated Predictions

After the baseline predictions were produced, the algorithm was modified to take the order in which each researcher joined each university into account. These modifications involved

replacing the Jaccard index with a different measure of how similar two researchers are. The first of these measures is the number of universities for which two researchers had an identical career progression. For example, if researcher A attended Stanford, Trinity College, N.Y.U. and M.I.T. in that order, while researcher B attended Stanford, Trinity College, University of Paris and M.I.T. in that order, then the similarity score between them would be 2 because the first and second universities they attended were the same but the third breaks the streak, even though the fourth is also the same. A second measure takes order into account but less strictly. If researcher C attended Stanford, University of Paris and M.I.T. in that order then the similarity between them and researcher B would be 3 as they attended 3 universities in the same order, even though there is an extra university in the middle. Both of these measured were compared, again using a k-nearest neighbours algorithm. For each researcher the first x universities they attended were used as training data, to assess their similarities to other researchers. Different values of x were tested to evaluate the trade-off between having more training data versus having less future universities that can potentially show up in the list of 10. Then once the k nearest neighbours were found and the list of the 10 most popular universities among them was created, the list was assessed under three metrics: Did the list contain any of the universities that the researcher would go on to work at, how many of them were contained in the list and what percentage of all of that researcher's future universities are contained in that list. For example, if a researcher will go on to work at 5 more universities and 2 are in the list of 10 possible future universities then the values of the previous metrics are yes, 2 and 40% respectively. Normally classification problems such as this are assessed using values such as accuracy, precision or recall. These metrics could be applied to this problem, but the metrics described above better describe the usefulness of the predictions made.

4 Machine Learning and Time Series Background

This section explains machine learning and time series techniques references in Section 3. If you are familiar with these techniques, there is no need for you to read this section.

4.1 Linear Regression

Linear regression is one of the most common machine learning algorithms. As its name suggests, it seeks to find a linear relationship between a set of features and a target variable. The equation representing this relationship is $h_\theta(x) = \theta^T x$, where x represents the vector of inputs, θ represents the vector of coefficients and $h_\theta(x)$ represents the prediction based on these inputs. To find the optimal values for the θ vector, it is first initialized with random values. These values are changed in each iteration to a value that makes the result of the cost function $J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^i\right) - y^i\right)^2$ smaller. To decide how much to modify these values, a gradient descent algorithm is used. For each value of the coefficient vector θ_j, the differential

of the cost function is calculated and subtracted from θ_j. This is described by the equations $\alpha \frac{\partial}{\partial \theta_j} J(\theta)$, where α is a pre-defined learning rate. When α is high, the values in θ change quickly but may skip over the optimal value which can increase the time taken to train the algorithm. It is important to note that gradient descent will converge to a local minimum, which may not be the same as the global optimal values for θ.

4.2 Random Forest

Random Forest is another machine learning algorithm. It makes use of multiple decision trees to form a hypothesis. Decision trees are one of the simplest machine learning methods but are prone to overfitting (fitting data too close to outlier instances) when they grow too deep. The Random Forest algorithm seeks to reduce this tendency to overfit data by implementing a modified form of bootstrap aggregation. This process involves training multiple decision trees on random subsets of the training data. These random subsets are extracted with replacement, meaning that multiple trees may be trained on subsets containing the same instances. To train a decision tree, for each node in the tree a condition is chosen that 'best' splits the dataset being used to train the tree. There are multiple algorithms for deciding the best split but all of them try to find a condition which is true for half of the data and false for the other half. When making predictions, a prediction is taken from each of the decision trees. These individual predictions are combined to form the final prediction, usually by averaging them for regression problems or returning the prediction that appears most often for classification problems. In addition to this standard bootstrap aggregation, Random Forest also only trains each decision tree using a random subset of features. This modification of the original bootstrap aggregation algorithm helps to avoid correlation of the decision trees where some of the features being trained on are much stronger predictors than the others.

4.3 K-Nearest Neighbours

K-Nearest Neighbours compares the features of the input feature vector to the features of every other instance in order to find the k most similar instances to that instance. Usually Euclidean distance is used to calculate the similarity between instances, with more similar instances having lower distances, but other measures can be used. Similar to random forest, the prediction from k-nearest neighbours is a combination of the target variables from the k most similar instances, usually using either the average of the values or the most common value. To test how well Linear Regression and Random Forest perform on different datasets, the dataset is split into 2 sections, training and test data. The algorithms are trained using the processes described above, and then predictions are made using the test data. These predictions are then assessed using a metric appropriate to the problem. For K-Nearest Neighbours, there is no

separation of training and test data, unlike the previous 2 algorithms, as each instance is compared to every other instance in the dataset.

Scikit-learn provides multiple versions of the k-nearest neighbours implementation: Ball Tree, K-Dimensional (KD) tree and brute force. All three of them were compared for these initial baseline tests. Brute force simply implements the algorithm as described above, attempting to find the k nearest neighbours based on the Minkowski distance between each feature vector. Minkowski distance is a hybrid between Euclidean and Manhattan distance measures. The other 2 variants make use of the training and test data split described above to improve the run time of the algorithm. The KD tree algorithm takes all the instances in the training dataset and orders them based on the feature that has the highest variance. It splits this list on the median instance for that feature. The two sub-lists are then sorted split on the median of the feature of highest variance that has not yet been used. This process is repeated until a set depth is reached or the lists are of length k. If the algorithm runs out of features to sort by, it will repeat the same order as before. The algorithm then goes through the training dataset, assigning each point an appropriate leaf node and making a prediction based on the instances from the training set contained in that sub-list. The ball tree algorithm optimises k-nearest neighbours by creating a binary tree where each node defines a multidimensional hypersphere or ball. To create this tree, instances in the training dataset are split into two initial balls, with no point appearing in both balls. Both balls are split again and this process is repeated until the leaf nodes of the tree are balls containing *k* points, or a certain depth is reached. Once this process is completed, the algorithm uses the same process as the KD tree algorithm to make predictions for the training dataset.

4.4 ARIMA Models

An ARIMA model is a generalised version of an autoregressive moving average (ARMA) model. Both models can be fitted to timeseries data and used to predict future values. "Autoregressive" indicates that the predictions for are made based on previous values of that timeseries only. Similar to a linear regression, autoregressive models model future values of a variable y_t as a linear combination of past values of that variable of the form $c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + e_t$ where y_t is the value of y at year t, ϕ_p is the coefficient by which y_{t-p} will be multiplied, c is a constant, e_t is white noise and p is the number of previous timesteps analysed by the model. This is different to the algorithms explained previously, which are fitted to observations from multiple instances and their target outputs. For each prediction made by an ARIMA model, the model is only fit to the available data of that specific timeseries. This means that to predict citation counts for this project, ARIMA models will only be fitted to the citation count timeseries, ignoring the publication count and h-index. The "Moving Average" part of the name indicates that once a model is fitted, future predictions are

based on the error between the model and the actual observed values rather than the values themselves. The above formula then becomes $c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + e_p y_{t-p}$, where e_t is the error between the model value and the observed value for year t combined with white noise.

Before combining these models to create an ARMA or ARIMA model, one must understand three central concepts regarding timeseries: stationarity, seasonality and differencing. A stationary timeseries is one where the value of the variable being measured is not dependent on the time at which it was measured and contains no overarching trends. A timeseries describing temperature in a specific location every 30 minutes for example would not be stationary because temperature will often peak around noon and then steadily decrease. A seasonal timeseries is one where patterns are consistently repeated independently of any trend in the data. A time series describing monthly sales of bicycles in a children's toy store may have sharp increases at the beginning of summer and in December even if the larger trend is decreasing. ARMA and ARIMA models can only be fitted to and make predictions on a stationary, non-seasonal time series.

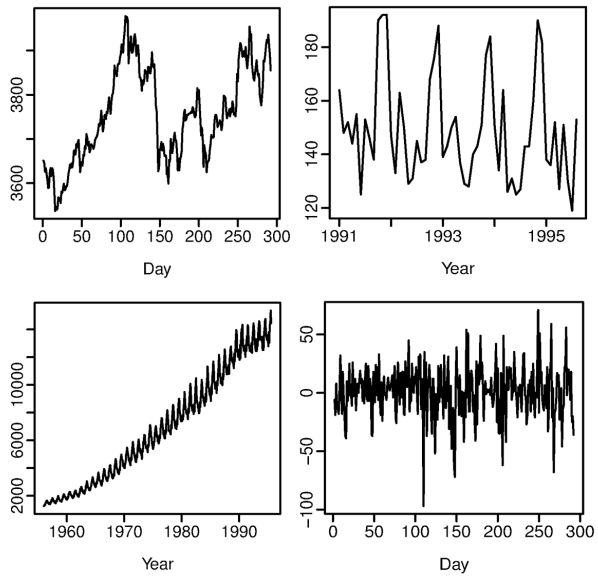

Figure 5: Graphs showing the daily value of the Dow Jones index (top left), monthly Australian beer production (top right), monthly Australian electricity production (bottom left) and the daily change in Dow Jones index (bottom right) [17]

Figure 5 shows four examples of time series. The daily Dow Jones index is not stationary because of the trends that appear in the timeseries. Monthly Australian beer production is seasonal due to its regular spikes. Monthly Australian electricity production is both seasonal and non-stationary due to the regular spikes and the overall upwards trend. The last graph, daily change in Dow Jones index, is both stationary and non-seasonal. It was created by substituting each value y_t in the original Dow Jones graph with $y_t - y_{t-1}$. This technique is

known as differencing and is often used to convert non-stationary or seasonal timeseries into stationary ones. If the difference from one day to the next can be predicted, then it is simple to add this predicted difference to the last observed value to produce a predicted value. Occasionally models will have to be differenced twice but differencing three or more times often distorts the data too much to create accurate predictions. Automatic differencing is the difference between ARIMA and ARMA algorithms. This is necessary as differencing can remove seasonality and trends from timeseries which is necessary to fit ARIMA models to them. The Statsmodels [28] library was used to implement ARIMA models for this project.

4.5 Multi-Layer Perceptron

A multi-layer perceptron is the simplest form of neural network. Inspired by the neural networks that exist in the human brain, a multi-layer perceptron consists of at least three layers of nodes: one input layer, one output layer and at least one hidden layer that sits between them. The input layer has one node for each feature that the network will be trained on, the output layer has as many nodes as there are target variables, and each hidden layer can have any number of nodes. Each node in the input layer is connected to every node in the first hidden layer, which are in turn connected to every node in the next hidden layer etc. The nodes in the input layer pass the inputted values to the next layer. Each node in the hidden layer then passes a value on to the nodes in the next layer. This value is determined by the value of the activation function of the neural network applied to the sum of all inputs into the node. Popular activation functions include Sigmoid and the Rectified Linear Unit function (ReLU). The formula for Sigmoid is $\sigma(z) = \frac{1}{1+e^{-z}}$ which approximates biological neurons. The formula for ReLU is simply $\text{ReLU}(z) = \max(0, z)$. To train a neural network, the weights assigned to each link are changed based on the error between the output value of the output layer and the observed value that corresponds to the provided inputs. This is done through backpropagation. Similar to the linear regression algorithm, the backpropagation process involves using a gradient descent to minimise the cost function $\varepsilon(n) = \frac{1}{2}\sum_j e_j^2(n)$ where $e(n)$ is the target value minus the predicted value. Each link is updated by the amount determined by the differential of this cost function, $-\alpha \frac{\partial \varepsilon(n)}{\partial w_{j,i}}$, where $w_{j,i}$ is the previous link weight between nodes j and i. This process is iterated until the weights of the links converge to a local minimum or a set number of iterations is reached. The Scikit-Learn library [27] was used to implement the multi-layer perceptron for this project.

4.6 LSTM

The final algorithm, a recurrent neural network (RNN), is a more complex type of neural network based on long short-term memory blocks (LSTMs). An LSTM is so named because it specialises in short term memory that is stored for a long time. In other words, it can remember

short patterns in a time series that may not be relevant to predictions until much further on in the time series. This makes it especially useful for predictive text models and speech recognition. Timeseries do not have to be stationary to be analysed by LSTMs but some non-stationary timeseries can much larger amounts of time and space to train than other machine learning algorithms. The LSTM based RNN for this project was implemented using the Keras [29] library.

4.7 K-Folds Cross Validation

K-folds cross validation is a way of evaluating the performance of machine learning algorithms. First the dataset is split into k sections or 'folds'. These instances assigned to each fold can be random or sequential. The machine learning algorithm is then trained on $k-1$ folds, with the remaining fold used as test data, with its performance assessed by a relevant metric such as RMSE etc. This process is repeated k times, ensuring each fold is used as test data once. The performance of the algorithm on each combination of folds is averaged to return the final value, similar to the bootstrap aggregating process used in the random forest algorithm. This averaging of multiple values helps to reduce the impact of outliers in the dataset worsening performance, or coincidental combinations of training and test data giving better predictions than the algorithm would normally. The effects of both of these conditions can also be decreased by increasing the size of the dataset, and for that reason K-folds cross validation is usually only used on smaller datasets. 10 fold cross validation was used with the algorithms described above.

4.8 RMSE and MAPE

The two metrics used to analyse the performance of many algorithms in this project are Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). RMSE is one of the most common metrics used for analysing machine learning algorithms because it returns the sample standard deviation of the differences between predicted values and observed values. This gives an easily intuitable measure of how good or bad a range of predictions are. For example, if an algorithm was designed to predict the rating that a person would give a movie out of 10, and it made 100 predictions for 100 different person-movie combinations with an RMSE of 1.7, that means that any future predictions are likely to be within 1.7 of the actual rating. This also holds for predicting citation counts. As the name suggests, RMSE returns the square root of the sum of the squares of the absolute difference between the predicted values and observed values. The formula for RMSE is $\mathrm{RMSE}\left(\hat{\theta}\right) = \sqrt{\sum((\hat{\theta}-\theta)^2)}$ where $\hat{\theta}$ is the vector of predicted values and θ is the vector of observed values.

RMSE is most useful when all observed values fall within a relatively narrow range as in the example above. For citation counts, where the observed values for citations received in a year can range from 0 to over a thousand, a single RMSE value may not necessarily be representative of the true performance of the algorithm being tested. For example, consider 6 hypothetical researchers who in their 10th year received 15, 23, 36, 204, 453, and 572 citations. An algorithm that returned predicted values of 65, 73, 86, 154, 403, 522 would have an RMSE of 50 as all predictions were off by 50. While these predictions may be within the acceptable margin of error for the larger observed values, they are not for the smaller ones. To help account for this lack of nuance in the RMSE results, MAPE is also used to assess the performance of the algorithms tested. MAPE returns the average of the percentage errors between predicted and observed values, using the formula MAPE $(\hat{\theta}) = (\frac{1}{N}\sum\frac{|\theta-\hat{\theta}|}{|\theta|}) \times 100$ where θ and $\hat{\theta}$ again represent. To return the previous example, both have an absolute error of 40, but the former has a percentage error of 6.5% while the latter has a percentage error of 67%. This ability to make use of percentage error in place of absolute error has both advantages and disadvantages. As demonstrated, percentage error can show details that may be missed by absolute error, but it can also hide others. For problems like citation count with a hard-minimum value of 0 citations for any given year, an algorithm that consistently outputs predicted values less than the observed values will have a maximum MAPE of 100, whereas algorithms that over predict have no limit on how high the MAPE can grow. In addition, poor performance on low observed values may negatively affect the MAPE. A predicted value of 8 citations when the observed value is 2 will have a percentage error of 400%, even though the prediction is relatively good. Additionally, observed values of 0 will lead to division by 0 in the standard MAPE formula, so for these predictions a modified formula was used where MAPE $(\hat{\theta}) = (\frac{1}{N}\sum\frac{|\theta-\hat{\theta}|}{|Max(\theta,\ 1)|}) \times 100$, a modification which was also used by Bertsimas et al. [13].

Neither of these metrics are perfect, but when combined they provide a much more complete picture than they do alone. Using both MAPE and RMSE in conjunction with each other can cause some problems in assessing the performance of different algorithms, especially in cases where performance improves as measured by one metrics but stays the same or worsens as measured by another. Possible explanations for these scenarios are explained in Section 5.2.

5 Results and Discussion

This section is split into four parts. The first provides some details about the metric and employment history datasets that were assembled. The second deals with the results of predicting citation count 5 years after the first 5, 10 and 15 years. The third deals with the compound predictions over the same time periods, predicting citation count, publication count and h-index. The fourth and final section deals with the results of employment predictions. The

second and third parts are further split into three sections, one for each of the three time-frames examined (The first 5, first 10 and first 15 years of a researcher's career). Each section compares the baseline results for each time-frame with the results from the more sophisticated algorithms, as well as presenting an argument as to which algorithm and dataset variant provides the best predictions for that time period. All confidence intervals shown in the following tables and graphs represent 90% of values.

5.1 Dataset Analysis

To provide context for the following results, the datasets for each predicted value are described below. Figure 6, Figure 7 and Figure 8 describe the publication count, citation count and h-index datasets respectively, showing the mean value and the standard deviation of each metric in each year. In Figure 6, a trend can be seen where a researcher's yearly publication count increases for the first 10 years, then plateaus for the next 20 years before beginning a downward trend late in their career. However, the large standard deviations in all 3 graphs imply that there is no typical career, with a wide range of values represented at all stages of an academic career. Each graph also shows the percentage of researchers in the dataset whose careers are longer than a certain year. For example, 80% of researchers in the dataset published their first document at least 11 years ago, while 40% first published at least 19 years ago. The decrease in sample size towards later years leads to some erratic behaviour in the graphs. For example, the sharp spike in citation count in year 38 of Figure 7, or the decrease and following increase in h-index after year 41 of Figure 8. This lack of researchers in later years also leads to the phenomenon of the average h-index decreasing at points, even though a researcher's individual h-index can never decrease. A researcher may have a relatively high h-index, increasing the average, but for years where there is no data for that researcher (because those years have not yet happened), the average h-index can decrease if the h-indexes of other researchers do not increase by enough to compensate for this loss.

The functionality of the Scopus API meant that while year by year document and citation count could be found for each researcher using only 1 API request per researcher, h-index required 1 API request per document. The limit of 20,000 requests per week imposed by Scopus made it impossible to acquire year by year h-index figures for every researcher in the list. The figures below are only based on the 451 researchers for whom all 3 metrics are available, although the baseline predictions described in Section 3.4 make use of the larger citations dataset which includes the details of up to 2417 researchers. The average, maximum and minimum values and the standard deviation for each value predicted in the following Section 5.2 are shown in the following figures and tables.

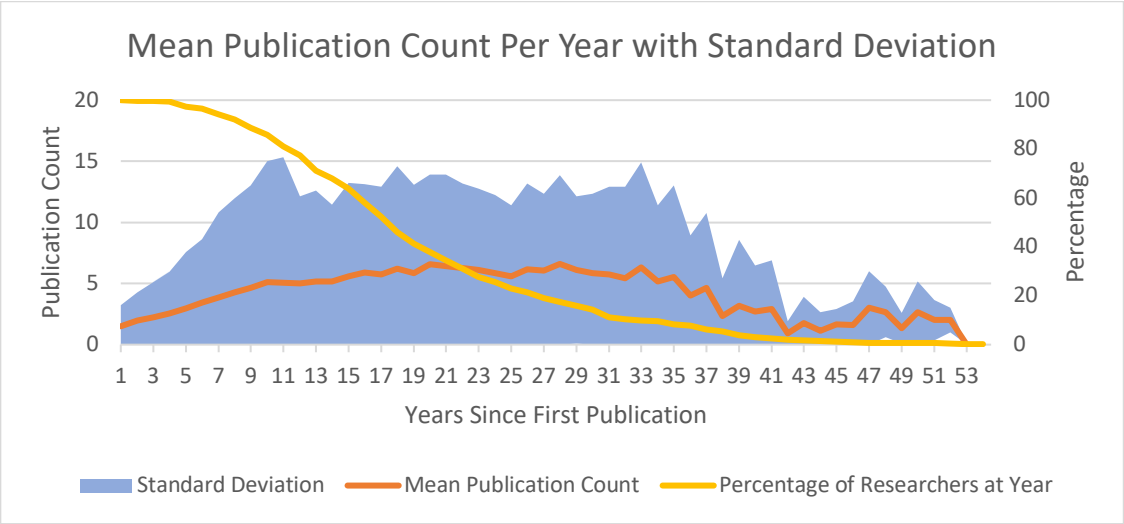

Figure 6: The publication count dataset

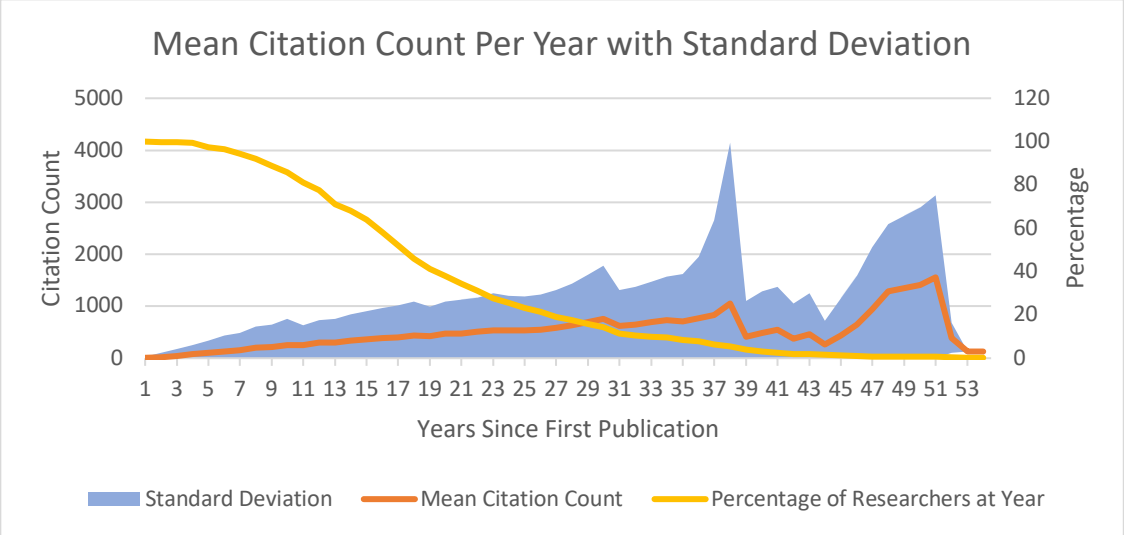

Figure 7: The citation count dataset

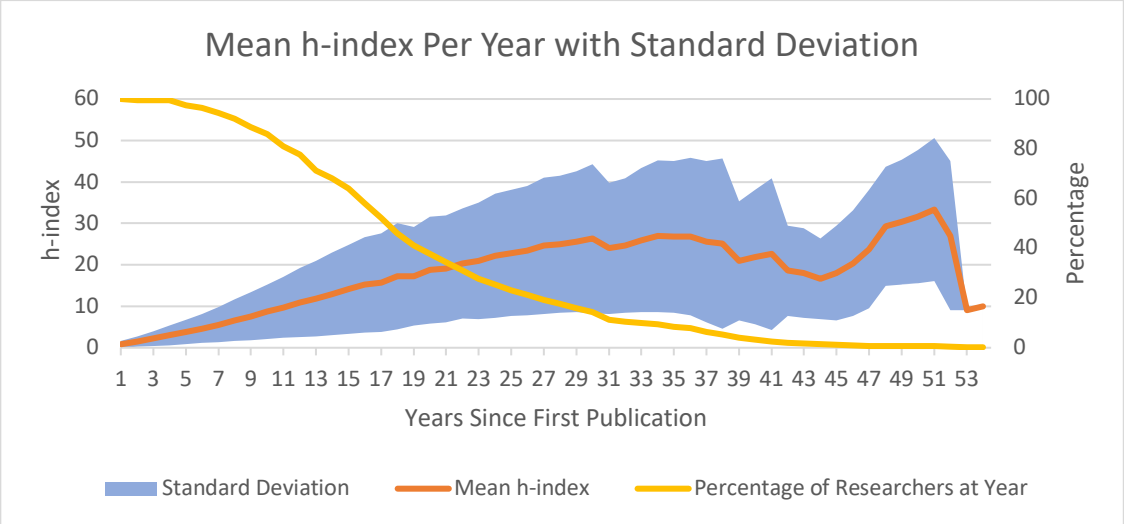

Figure 8: The h-index dataset

Publication Count	Average	Minimum	Maximum	Standard Deviation
Year 6	3.5	0	67	5.1
Year 7	3.8	0	71	7
Year 8	4.3	0	118	7.7
Year 9	4.7	0	128	8.4
Year 10	5.1	0	138	9.9
Year 11	5.1	0	161	10.3
Year 12	5	0	146	7.1
Year 13	5.1	0	61	7.5
Year 14	5.2	0	87	6.3
Year 15	5.6	0	72	7.6
Year 16	5.9	0	118	7.2
Year 17	5.7	0	145	7.2
Year 18	6.2	0	179	8.4
Year 19	5.9	0	223	7.2
Year 20	6.6	0	274	7.3

Table 1: Average, maximum and minimum values and standard deviation of publication counts in years 6 through 20

Citation Count	Average	Minimum	Maximum	Standard Deviation
Year 6	134	0	3222	306
Year 7	156	0	4486	334
Year 8	197	0	5631	410
Year 9	216	0	6249	435
Year 10	256	0	7058	507
Year 11	254	0	2924	379
Year 12	296	0	3545	438
Year 13	305	0	4217	448
Year 14	344	0	4927	505
Year 15	364	0	5421	542
Year 16	385	0	5870	577
Year 17	397	0	6396	615
Year 18	440	0	6512	645
Year 19	422	0	6455	570
Year 20	469	0	6794	625

Table 2: Average, maximum and minimum values and standard deviation of citation counts in years 6 through 20

h-index	Average	Minimum	Maximum	Standard Deviation
Year 6	4.6	0	26	3.5
Year 7	5.6	0	34	4.3
Year 8	6.6	0	37	5
Year 9	7.6	0	42	5.8
Year 10	8.7	0	45	6.6
Year 11	9.7	0	48	7.4
Year 12	10.9	0	50	8.3
Year 13	11.8	0	53	9.1
Year 14	13	0	56	9.9
Year 15	14.1	0	57	10.7
Year 16	15.2	0	65	11.5
Year 17	15.7	0	72	11.9
Year 18	17.2	0	75	12.8
Year 19	17.3	0	61	11.9
Year 20	18.7	0	64	12.9

Table 3: Average, maximum and minimum values and standard deviation of publication counts in years 6 through 20

Figure 9 shows the frequency distribution of the universities in the employment dataset. 49% of the universities appear just once in the dataset, with over 80% appearing 5 or less times. This implies that there is a wide range of universities represented. This wide spread is also indicated in Figure 10, which shows the frequency of the 25 most common universities. The most attended university in the dataset, Stanford, appears 192 times, whereas the 25th most attended, Tsinghua University, appears only 38 times, an 80% drop. Figure 11 shows the frequency of the number of universities a researcher has worked at. The orange line shows the percentage of researchers who have worked at more universities than the value on the x axis. 25% of the researchers in the dataset have worked at only 1 university, while 81% have worked at 5 or less. The mean number of universities worked at is 4.1, with a standard deviation of 7.4.

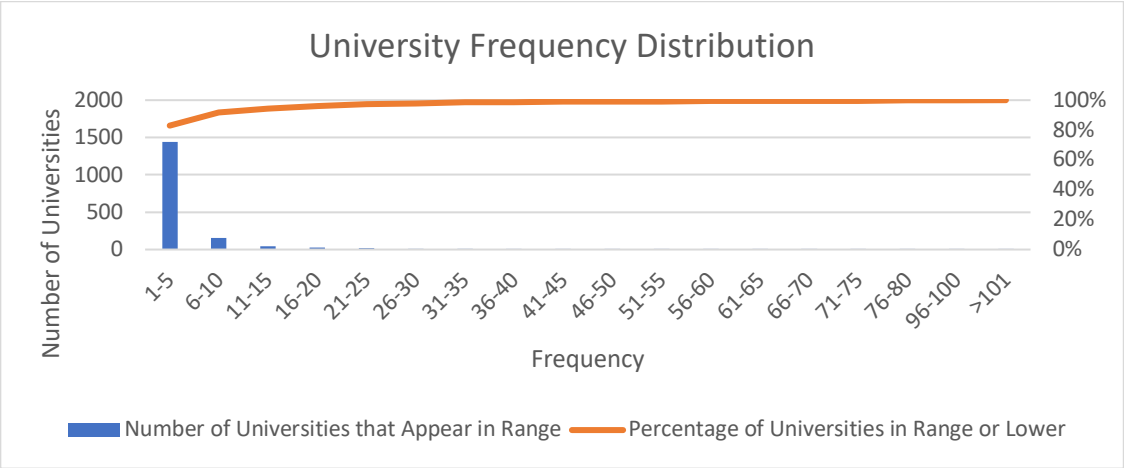

Figure 9: University Frequency Distribution

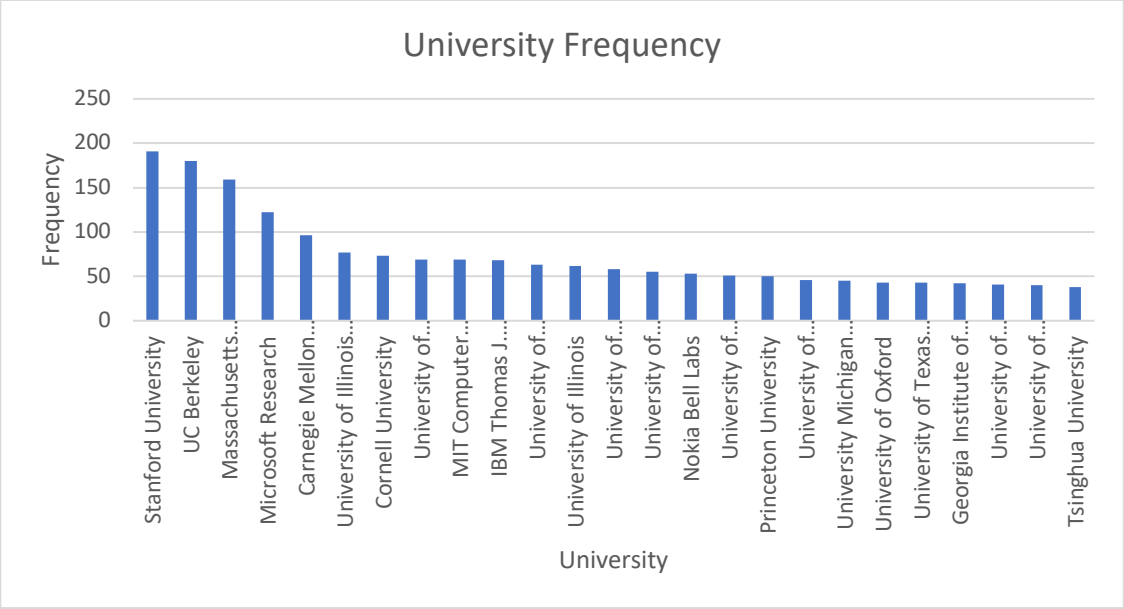

Figure 10: Frequency of the 25 most common universities

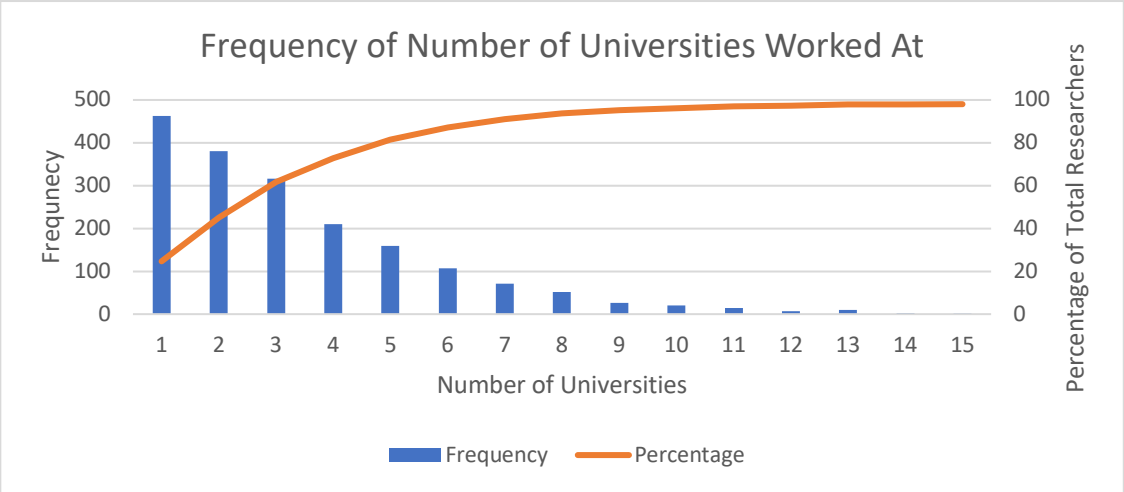

Figure 11: Frequency of number of universities worked at

5.2 Single Metric Prediction

While gathering these results, two algorithms were ruled out as they were found to be unsuitable for the stated research problem. ARIMA models could not be fit to 37% of citation histories as even after differencing them twice, they were still found to be non-stationary. The Recurrent Neural Network based on LSTMs consistently failed to converge when trained on the multivariate datasets. This is likely because LSTMs are designed to identify sequences based on the historical context of inputs, while the results below show that many other algorithms attain the best results when only looking at more recent input years. While results for some researchers could be produced by both models, the high failure rates meant that they were fundamentally unsuited to the problem. Even if MAPE and RMSE values were generated based on their predictions, they would not be comparable to those of other algorithms due to these failure rates.

5.2.1 Predictions from 5 to 10 years

Table 4 and Figure 12 show the MAPE and RMSE values for the baseline predictions of total citation count in the 10th year of a researcher's career based on the citations attained in each of their first five years. K-Nearest Neighbours outperforms both Random Forest and Linear Regression, with the ball tree optimisation providing the best results. This weighted average provided the best results of all algorithms tested. Its MAPE and RMSE of 199 and 1325 are both better than the 211 and 1338 attained by the uniform weighting variant. They are also better than the distance weighting variant of the KD tree algorithm, which attained scores of 206 and 1334 for MAPE and RMSE respectively. As illustrated in Figure 12, all predictions have very wide confidence intervals, a trend that appears in most results.

Baseline Predictions	**MAPE**	**Confidence Interval**	**RMSE**	**Confidence Interval**
KNN ball tree/distance	199	±116	1325	±590
KNN ball tree/uniform	212	±151	1339	±529
KNN brute/distance	254	±181	1332	±522
KNN brute/uniform	256	±175	1340	±556
KNN KD tree/distance	207	±126	1335	±546
KNN KD tree/uniform	218	±128	1342	±570
Random Forest	267	±143	1400	±525
Linear regression	1496	±795	1460	±548

Table 4: MAPE and RMSE values for baseline predictions of citation count in year 10 based on citation count of first 5 years

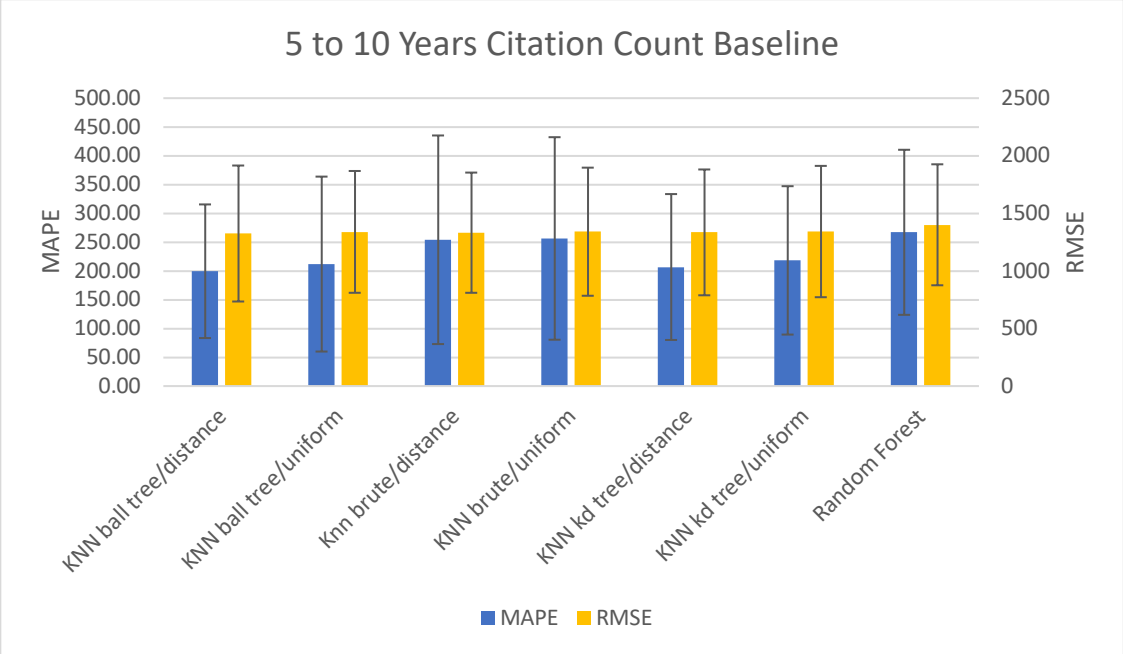

Figure 12: MAPE and RMSE values for baseline predictions of citation count in year 10 based on citation count of first 5 years (Linear Regression not included as MAPE was over 5 times larger than next highest MAPE, making comparisons between other algorithms difficult).

Table 5 and Figure 13 compares the best performing algorithms from each version of the datasets described in Section 3.4.1.2 with the best performing algorithm from the baseline predictions. These results show that some of the algorithms perform well as measured by MAPE and poorly as measured by RMSE or vice versa. The best algorithms were therefore chosen were not because they had the lowest value for MAPE or RMSE, but rather because they achieved relatively good scores in both compared to other algorithms tested on the same dataset. All algorithms except one are an improvement over the best baseline prediction as measured by MAPE. The multi-layer perceptron algorithm with a one year lag predicting the difference between a researchers 5 and 10 year total citation count has the best MAPE, 119, an improvement of 40% compared to the MAPE of the baseline, 199. However, this algorithm is the worst of those using the multivariate dataset as measured by RMSE. Its RMSE of 876 is 20% higher than the best RMSE of 721, achieved by a multi-layer perceptron with a 4 year lag running on the cumulative version of the dataset and again predicting the difference between a researcher's 5 and 10 year citation count. Despite this, when assessing algorithms by both metrics it is one of the best performing thanks to its low MAPE. It and the K-Nearest Neighbours algorithm running on the individual years dataset are the two standout algorithms, but their contradictory values for MAPE and RMSE make it difficult to choose the best algorithm. The average number of citations attained by a researcher by year 10 in this dataset is 1045, with a standard deviation of 2172. These values indicate that even the lowest RMSE attained by these algorithms of 721 is not a very good value, and these predictions would have to be improved before they could be said to be accurate.

Best Predictions	MAPE	Confidence Interval	RMSE	Confidence Interval
Individual Years KNN 5 year Lag	140	75; 220	828	434; 1129
Cumulative Years MLP 2 Year Lag	166	93; 238	769	439; 1322
Individual Difference MLP 1 Year Lag	119	67; 175	876	589; 1304
Cumulative Difference MLP 4 year Lag	202	130; 298	721	297; 948
Baseline KNN Ball Tree	199	83; 315	1325	734; 1915

Table 5: The MAPE and RMSE of the best performing algorithms and their best performing lag sizes for each dataset

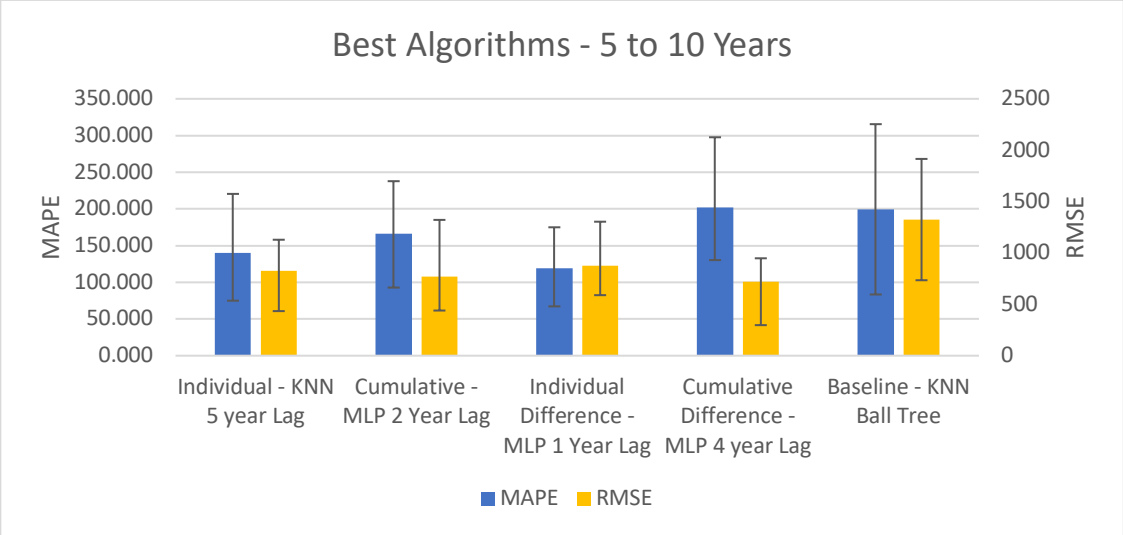

Figure 13: The MAPE and RMSE of the best performing algorithms and their best performing lag sizes for each dataset

5.2.2 Predictions from 10 to 15 years

Table 6 and Figure 14 show the MAPE and RMSE scores of the baseline predictions for a researcher's total citation count after 15 years based on the citations they received in the first 10 years of their career. Comparing these results to the corresponding results for predicting citation count in the 10th year reveals several insights. The scores received by the K-Nearest Neighbours variants for these predictions are much closer than for the previous set. These predictions have MAPE values between 78 and 80, an increase of just over 1% from lowest to highest, whereas the previous set ranged from 199 to 256, an increase of 29%. The RMSE values are similarly close, ranging from 1957 to 2053, also a 1% increase. These close values coupled with the large confidence intervals indicate that the variant used does not significantly affect the results. These MAPE values are also significantly lower than those of the 10 year predictions. This makes sense as there are twice as many years to train each algorithm on, allowing for potentially more accurate predictions. The RMSE values are much higher but this

too can be expected as the average total citation count of researcher's in the 10th year of their career is 1045, while in their 15th year this number almost doubles to 2079. That the RMSE values for all baseline algorithms are near this point means that the predictions are not much better than random guessing centred on the average value.

Baseline Predictions	**MAPE**	**Confidence Interval**	**RMSE**	**Confidence Interval**
KNN ball tree/distance	79	±28	1973	±1413
KNN ball tree/uniform	79	±31	2017	±1382
KNN brute/distance	77	±34	2053	±1328
KNN brute/uniform	79	±26	2022	±1382
KNN KD tree/distance	78	±29	1957	±1449
KNN KD tree/uniform	80	±27	2006	±1387
Random Forest	94	±39	2147	±1516
Linear regression	842	±632	2087	±938

Table 6: MAPE and RMSE values for baseline predictions of citation count in year 15 based on citation count of first 10 years

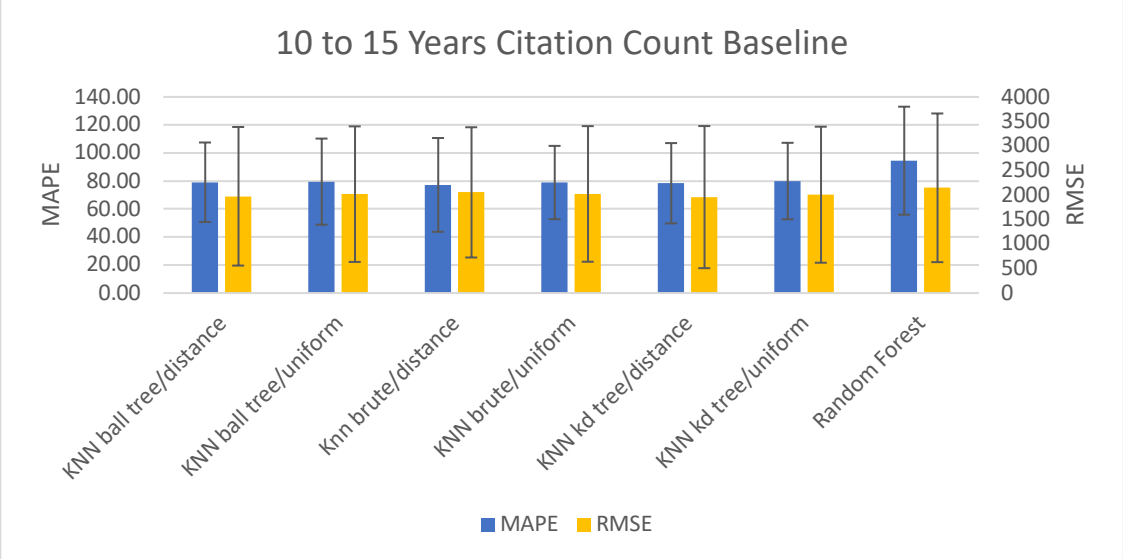

Figure 14: MAPE and RMSE values for baseline predictions of citation count in year 15 based on citation count of first 10 years (Linear Regression not included as MAPE was almost 9 times larger than next highest MAPE, making comparisons between other algorithms difficult).

Even though there were multiple algorithms with very similar scores, K-Nearest Neighbours with the KD tree optimisation was chosen to be compared with the multivariate algorithms as it had the lowest scores in both MAPE and RMSE. Table 7 and Figure 15 show the RMSE and MAPE values with confidence intervals for the best performing algorithm on each dataset

variant compared to the K-Nearest Neighbours algorithm with this algorithm. The MAPE values of the multivariate algorithms, while a large improvement over their 10 year counterparts, perform similarly to the best baseline algorithm. Conversely, huge improvements can be seen as measured by RMSE, indicating again that the use of multiple input features improves performance. The best MAPE, achieved by a Multi-Layer Perceptron with a seven year lag predicting the difference between a researchers 10 and 15 year citation count based on the citations achieved in each individual year, is only a 4% improvement over the baseline algorithm, 78 to 76, and has a much wider confidence interval. The best RMSE however, achieved by the same algorithm, is 682, a 65% decrease over the baseline RMSE of 1957. A correlation between more input years and lower MAPE values was noted in the previous paragraph, but the best performing algorithms in fact operate on relatively small lag sizes, even though the lowest MAPE is produced by a Multi-Layer Perceptron running on a 7 year lag. This would indicate that the number of input years does not matter as much as how late in a researcher's career these years are, with data from more recent years being a much stronger predictor than data from earlier years.

Best Predictions	**MAPE**	**Confidence Interval**	**RMSE**	**Confidence Interval**
Individual Years MLP 6 Year Lag	84	50; 123	683	448; 861
Cumulative Years MLP 3 Year Lag	80	45; 124	828	549; 1305
Individual Difference MLP 7 Year Lag	76	39; 128	682	516; 811
Cumulative Difference MLP 3 Year Lag	77	53; 122	734	550; 964
Baseline KNN KD Tree Distance	78	50; 107	1957	508; 3406

Table 7: The MAPE and RMSE with confidence intervals of the best performing algorithms and their best performing lag sizes for each dataset

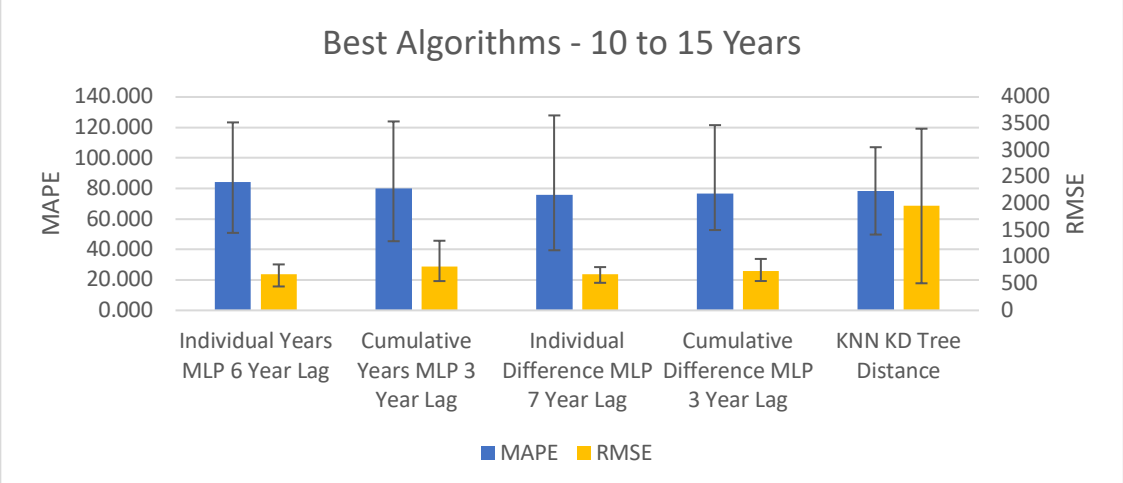

Figure 15: The MAPE and RMSE with confidence intervals of the best performing algorithms and their best performing lag sizes for each dataset

While it is difficult to compare RMSE scores from the 10 and 15 year predictions directly, some comparison is possible. The best RMSE for the 15 year predictions of 682 is 32% of the average total citation count of 2079 in year 15, while the best RMSE of the 10 year predictions, 721, is 68% of the average total citation count in year 10, 1045. This relative decrease indicates better performance, something that is echoed by the improvement in MAPE scores from 119 for year 10 predictions to 76 for year 15 predictions. Another trend from the 10 year results that is seen here is the strong performance of the Multi-Layer Perceptron algorithm. It was the best performing algorithm on each dataset variant for the 15 year predictions, as it was for three out of the four variants of the 10 year predictions.

5.2.3 Predictions from 15 to 20 years

Table 8 and Figure 16 show the baseline predictions for a researcher's total citation count in the 20th year of their career based on the citations received in each of the first 15 years. Similar trends to the previous two sets of baseline predictions can be seen again here. The K-Nearest Neighbours algorithms perform better than Random Forest and Linear Regression, with the distance weighted variants outperforming their uniformly weighted counterparts. The best performing algorithm is the brute force, distance weighted K-Nearest Neighbours algorithm with an MAPE and RMSE of 59 and 2642 respectively. This is the best value of RMSE by 114, a margin which makes up for having an MAPE 3 higher than the lowest. The decrease in MAPE from all algorithms again shows a correlation between more training years and lower MAPE scores. Wide confidence intervals are also a feature of these results, as they have been in both sets of previous baseline predictions.

Baseline Predictions	**MAPE**	**Confidence Interval**	**RMSE**	**Confidence Interval**
KNN ball tree/distance	56.	±22	2756	±1870
KNN ball tree/uniform	59	±23	2727	±1553
KNN brute/distance	59	±27	2642	±1716
KNN brute/uniform	60	±26	2713	±1753
KNN KD tree/distance	60	±26	2681	±1739
KNN KD tree/uniform	60	±25	2710	±1852
Random Forest	108	±118	2692	±1225
Linear regression	693	±724	2684	±1443

Table 8: MAPE and RMSE values for baseline predictions of citation count in year 20 based on citation count of first 15 years

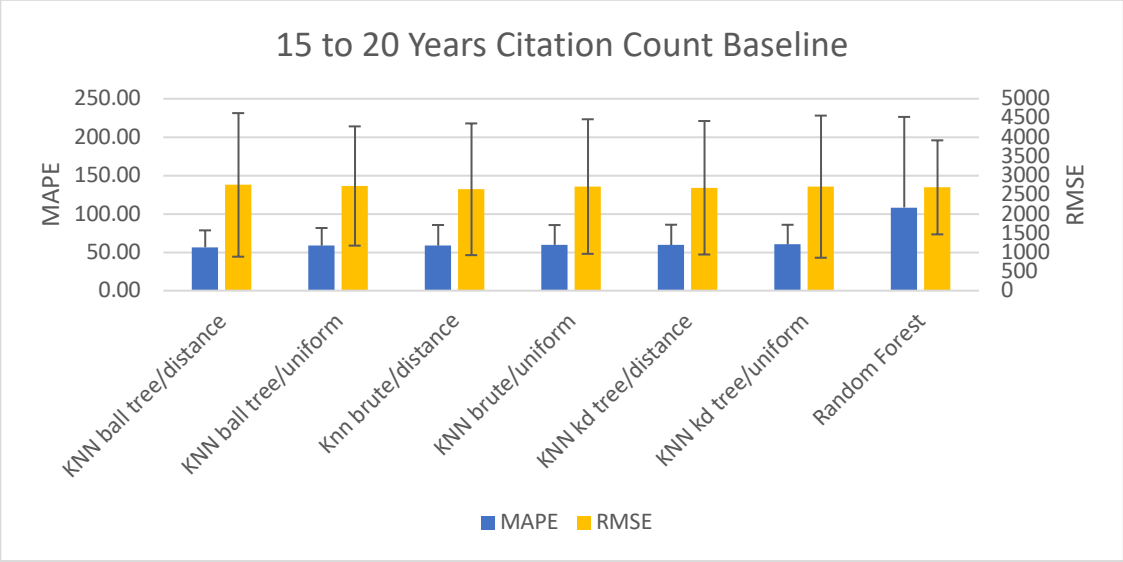

Figure 16: MAPE and RMSE values for baseline predictions of citation count in year 15 based on citation count of first 10 years (Linear Regression not included as MAPE was almost 9 times larger than next highest MAPE, making comparisons between other algorithms difficult).

Table 9 and Figure 17 show the MAPE and RMSE values of the best performing algorithms for each dataset variant. In contrast to the other two time periods examined, Linear Regression is the best performing algorithm in three out of four variants, indicating that the relationship between the first 15 years of a researcher's career and their total citation count in their 20th year can be much easier defined by a linear model than earlier time periods. This implies that after 15 years a researcher's career has become very well defined, and the number of citations they achieve per year is unlikely to change significantly. The MAPE of all algorithms is better than the best performing algorithm for the 15 year predictions, and most of the multivariate algorithms outperform the best baseline algorithm. The trend of small lag sizes receiving better MAPE scores as seen in the 15 year predictions is again seen here, with all the best performing algorithms running on lags of less than 4 years even though all sizes up to 15 were tested. The Linear Regression algorithm, like the best performing algorithms of the other two time periods, is running on the individual years dataset and predicting the difference between the total citation count at the cut-off point and the total citation count 5 year later. It has the lowest RMSE score, 934, and the second lowest MAPE score, 50. Only the Multi-Layer Perceptron, running on the same dataset but predicting just the total citation count and not the difference, received a lower MAPE score, 39. The average total citations achieved by a researcher in their 20th year is 3495, with a standard deviation of 5733. This makes the best RMSE of 934 a relatively good prediction, only 26% of this average. This coupled with the relatively low

MAPE of 50% makes this the time period with the best results, although further improvement is obviously still possible.

Best Predictions	**MAPE**	**Confidence Interval**	**RMSE**	**Confidence Interval**
Individual Years MLP 2 Year Lag	39	21; 57	1012	793; 1450
Cumulative Years Lin. Reg. 3 Year Lag	61	34; 76	1090	745; 1340
Individual Difference Lin. Reg. 3 Year Lag	50	30; 66	934	487; 1108
Cumulative Difference Lin. Reg. 4 Year Lag	60	32; 95	948	682; 1152
KNN Brute Force Distance	59	31; 85	2642	926; 4358

Table 9: The MAPE and RMSE with confidence intervals of the best performing algorithms and their best performing lag sizes for each dataset

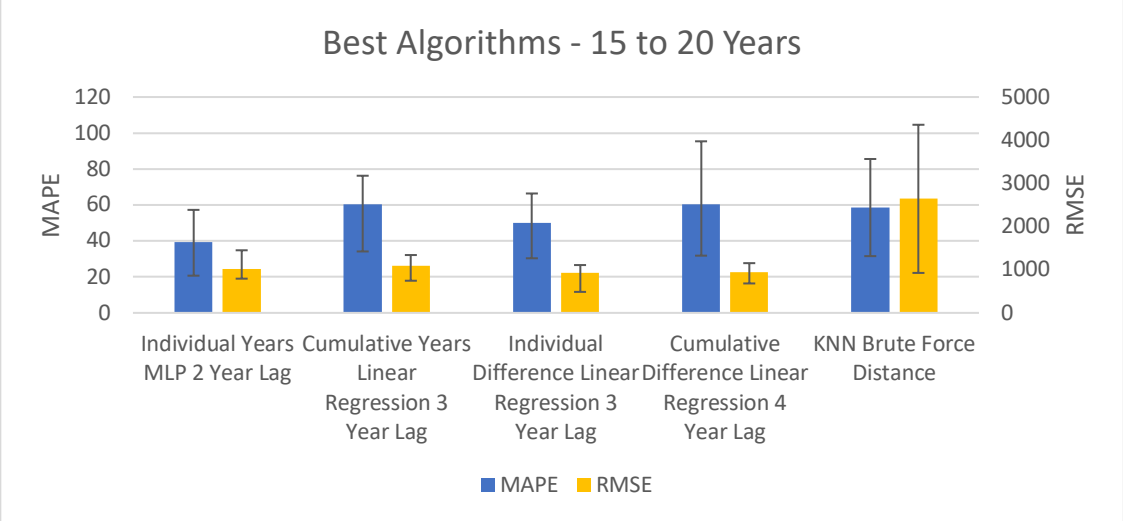

Figure 17: The MAPE and RMSE of the best performing algorithms and their best performing lag sizes for each dataset

Many of these results show that the introduction of new features improves performance and more sophisticated algorithms like multi-layer perceptrons can improve it even more, but the seemingly contradictory measurements of RMSE and MAPE in some cases make it difficult to conclusively decide which algorithm has the best performance. One explanation for this phenomenon comes from the problems with MAPE described in Section 4.8. An algorithm could artificially improve its MAPE by making better predictions for low numbers but worse predictions for high numbers, while keeping its RMSE the same or worsening it. Another explanation is that an algorithm that is biased towards predicting values less than the observed value can also increase MAPE when there is a natural floor on predictions, as there is for citation counts. A negative bias in scenarios like this means that MAPE cannot exceed 100, while a positive bias has no such limit. For example, if the observed values for similar input feature vectors were 5, 15, 350, 400, 600 and 20 and the corresponding predicted values were

100, 250, 370, 390, 580 and 200, clustered around the midpoint of these observed values, the MAPE would be 729 and the RMSE would be 128. However, if the observed values had a negative bias, for example 20, 50, 70, 80, 200 and 140 then the MAPE would decrease to 227 and the RMSE would increase to 252. The occurrence of this phenomenon may indicate that some algorithms are better suited for making predictions for researchers with low citation counts and other algorithms can better predict those with high citation counts. A combination of the results of different algorithms may then be useful to create more useful predictions.

5.3 Compound Metric Predictions

5.3.1 5 to 10 Years

The same four algorithms were used to initially predict the values of h-index, publication count and citation count one year into the future. These predicted values were then used as training data and used to predict the same values two years into the future, and the process was repeated for five years in total. Table 9 and Figure 18 show the RMSE values for the predicted h-index values of each year with the confidence intervals. MAPE is neither used here nor when assessing publication count prediction, as the small values make percentage errors difficult to interpret (A predicted value of 3 for an observed value of 2 results in a 50% percentage error). Interestingly, the algorithms that performed worse on the 5 to 10 year citation count single predictions perform better in the year by year h-index predictions. Table 9 and Figure 18 clearly show Linear Regression and Random Forest outperforming the Multi-Layer Perceptron and K-Nearest Neighbours algorithms to predict a researcher's h-index, with Random Forest's RMSE of 2.83 in year 10 a 46% improvement over the Multi-Layer Perceptron's score of 5.261. With an average of 8.7 a standard deviation of 6.6 in year 10, Random Forest's RMSE of 2.83 makes it a useful algorithm for predicting h-index.

h-index	**Year 6**	**Year 7**	**Year 8**	**Year 9**	**Year 10**
Multi-Layer Perceptron	1.1	1.8	2.7	4.3	5.3
Confidence Interval	0.0; 2.0	0.0; 3.0	0.0; 4.0	0.3; 7.0	0.8; 8.0
K-Nearest Neighbours	1.5	2.7	3.7	4.5	5.0
Confidence Interval	0.0; 3.0	0.0; 4.9	0.0; 6.0	0.0; 7.0	0.0; 8.0
Linear Regression	0.9	1.3	1.8	2.4	3.3
Confidence Interval	0.0; 1.0	0.0; 2.0	0.0; 3.0	0.0; 4.0	1.0; 6.0
Random Forest	0.8	1.2	1.8	2.3	2.8
Confidence Interval	0.0; 1.0	0.0; 2.0	0.0; 3.0	0.0; 4.0	0.0; 5.0

Table 9: RMSE values with confidence intervals for predicting the values of a researcher's h-index in each of the 5 years after their first 5 years.

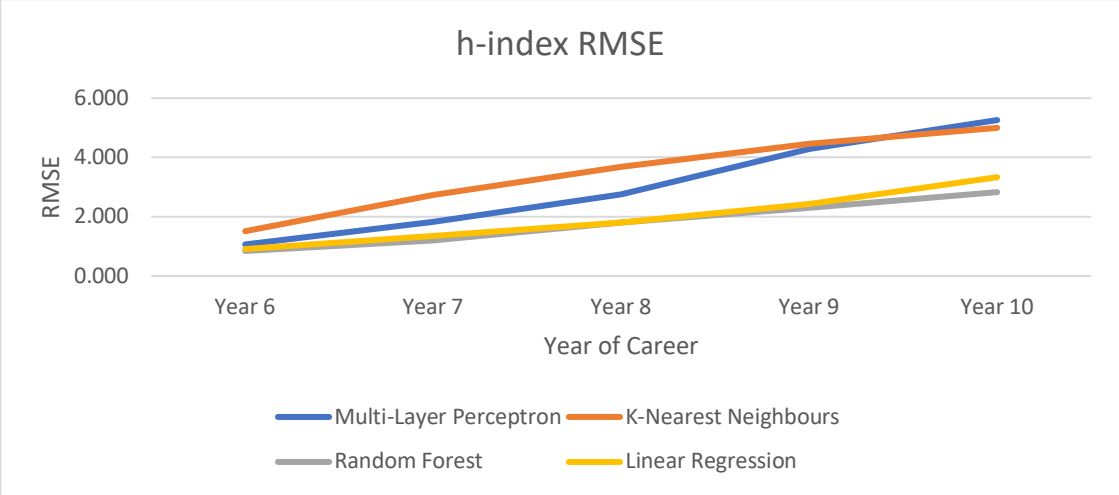

Figure 18: RMSE values for predicting the values of a researcher's h-index in each of the 5 years after their first 5 years.

Table 10 and Figure 19 show the RMSE values for predicting publication count in a similar manner. The Multi-Layer Perceptron algorithm performs much better in these predictions than it did for h-index, while K-Nearest Neighbours remains the worst performing algorithm. Random Forest has the lowest RMSE in each year with 4.8 in year 10 but after taking the wide confidence intervals into account it is impossible to say from this data alone which algorithm performs best at this task. In any case, with an average of 5.1 and standard deviation of 9.9 in year 10, none of the results can be said to accurately predict publication count.

Publication Count	**Year 6**	**Year 7**	**Year 8**	**Year 9**	**Year 10**
Multi-Layer Perceptron	4.0	4.3	5.2	6.1	5.8
Confidence Interval	1; 4.7	1.0; 5.0	1.0; 10.0	1; 11.6	1.0; 6.0
K-Nearest Neighbours	6.1	9.6	10.8	12.3	14.5
Confidence Interval	0.0; 11.7	0.0; 18.3	0.0; 20	1.0; 23.5	0.3; 27.5
Linear Regression	3.1	4.1	4.8	5.6	6.3
Confidence Interval	0.0; 6.0	0.0; 6.0	0.0; 7.0	0.0; 8.0	0.0; 9.0
Random Forest	2.7	3.5	4.2	4.4	4.9
Confidence Interval	0.0; 4.0	0.0.; 60.	0.0; 6.0	0.0; 8.0	0.0; 7.0

Table 10: RMSE values with confidence intervals for predicting the values of a researcher's publication count in each of the 5 years after their first 5 years.

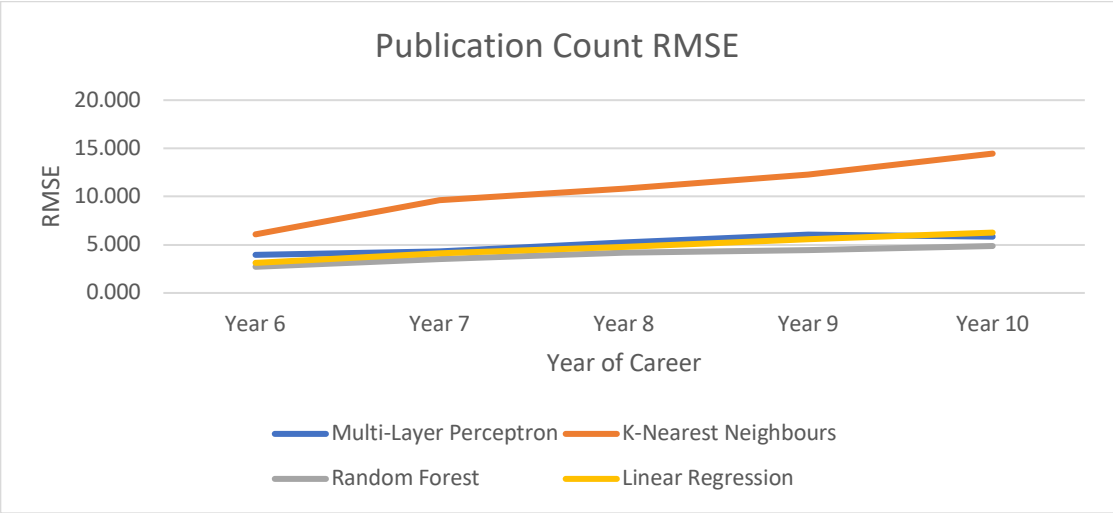

Figure 19: RMSE values for predicting the values of a researcher's publication count in each of the 5 years after their first 5 years.

Figure 20, Figure 21, Table 11 and
Table 12 show the RMSE and MAPE values for predicting year by year citation counts. As can be clearly seen, the four algorithms perform in essentially reverse order when measured by MAPE and RMSE, making it difficult to say conclusively which performs best. What is clear is that none of these algorithms can accurately predict citation count very well, with a best RMSE of 327 in year 10 when the average number of citations received in this year is 256 with a standard deviation of 507.

Citation Count - MAPE	**Year 6**	**Year 7**	**Year 8**	**Year 9**	**Year 10**
Multi-Layer Perceptron	138	221	343	484	583
Confidence Interval	9; 400	11; 800	15; 1100	19; 1418	20; 1391
K-Nearest Neighbours	60	88	95	95	130
Confidence Interval	2; 100	6; 125	7; 207	8; 207	18; 272
Linear Regression	113	203	273	378	501
Confidence Interval	5; 333	10; 508	11; 862	14; 944	19; 1048
Random Forest	87	138	198	257	312
Confidence Interval	5; 206	3; 243	9; 469	12; 593	12; 635

Table 11: MAPE values with confidence intervals for predicting the values of a researcher's citation count in each of the 5 years after their first 5 years.

Citation Count - RMSE	**Year 6**	**Year 7**	**Year 8**	**Year 9**	**Year 10**
Multi-Layer Perceptron	37	71	125	210	327
Confidence Interval	2; 40	4; 80	11; 128	17; 405	22; 631
K-Nearest Neighbours	161	253	348	396	474
Confidence Interval	1; 290	1; 463	1; 641	2; 739	3; 896
Linear Regression	73	136	196	274	364
Confidence Interval	2; 140	4; 259	5; 374	12; 525	17; 701
Random Forest	132	224	322	406	483
Confidence Interval	1; 241	1; 410	2.3; 598	4; 766	5; 919

Table 12: RMSE values with confidence intervals for predicting the values of a researcher's citation count in each of the 5 years after their first 5 years.

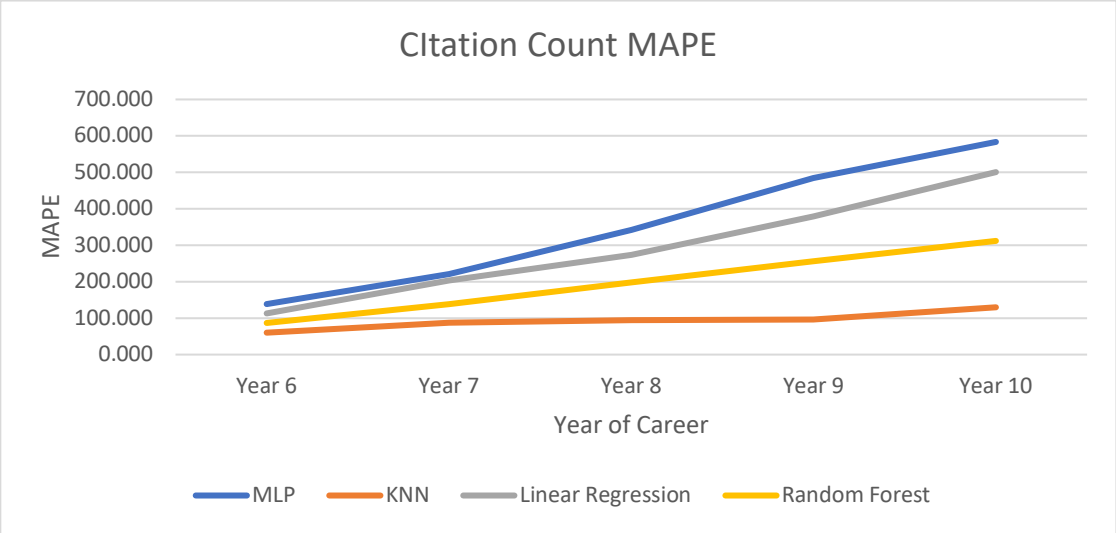

Figure 20: MAPE values for predicting the values of a researcher's citation count in each of the 5 years after their first 5 years.

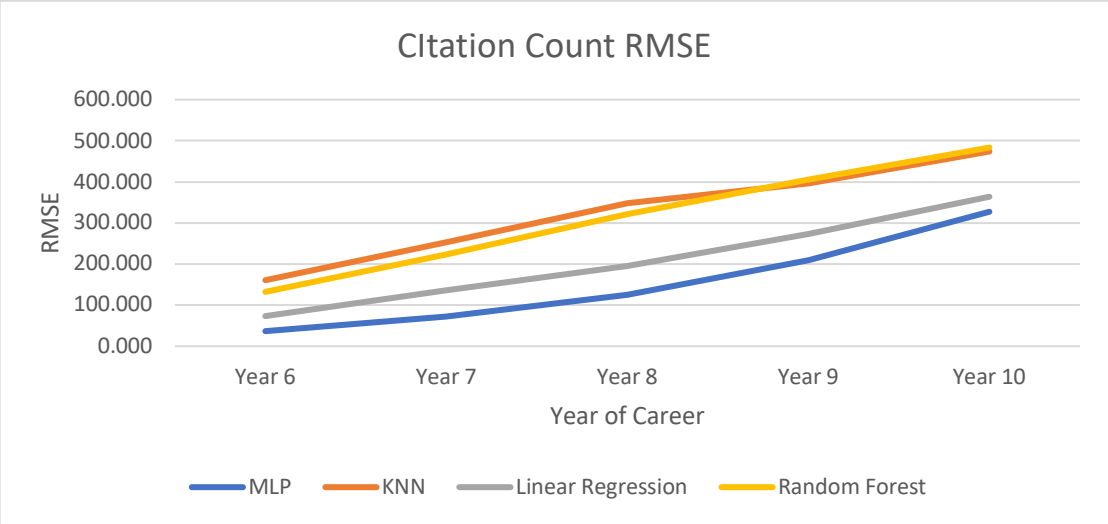

Figure 21: RMSE values for predicting the values of a researcher's citation count in each of the 5 years after their first 5 years.

These results show that the prediction of the citation count of a researcher in their 10th year is much better made in one prediction rather than making 5 compound predictions. The errors from each smaller prediction compound and skew the final results much more than in a single prediction. To get the total citations in year 10 from these year by year predictions, the results from each year would have to be summed, which would compound the errors of each individual year. One of best performing algorithms from the single predictions had an RMSE of 876 when predicting the total citation count in year 10. The best RMSE for predicting only the citations received in year 10 is 327, 37% of that. While it's possible that the errors from other years may be both positive and negative and cancel each other out, it is far more likely that they will compound and produce a worse prediction than the single predictions described in Section 5.2.

5.3.2 10 to 15 Years

Table 13 and Figure 22 show the RMSE scores for different algorithms predicting the value of a researcher's h-index each year between the 10th and 15th year of their publishing career. As was the case in for predicting the results between the 5th and 10th year, the Linear Regression and Random Forest algorithms perform the best of the four, but while they performed similarly in the previous set of predictions, here Random Forest performs significantly better than Linear regression, with an RMSE of 3.9 and a confidence interval of 0 to 6 in year 15 compared to an RMSE of 5.9 and confidence interval of 0 to 11.3 for Linear Regression. The average h-index of a researcher 15 years after they first publish is 14.1 with a standard deviation of 10.6, making an RMSE of 3.9 a reasonably good score.

h-index RMSE	**Year 11**	**Year 12**	**Year 13**	**Year 14**	**Year 15**
Multi-Layer Perceptron	4.9	5.1	4.9	6.1	7.9
Confidence Interval	0; 9.5	0.2; 5.1	1; 7.6	0.9; 8.8	1.0; 12.0
K-Nearest Neighbours	3.0	4.4	4.9	5.7	6.6
Confidence Interval	0.0; 4	0.0; 7.0	0; 8	0.0; 9.0	0.1; 12.0
Linear Regression	1.2	2.4	3.5	4.6	5.9
Confidence Interval	0.0; 2.0	0.0; 3.0	0.0; 6.7	0.0; 8.8	0; 11.3.0
Random Forest	1	1.6	2.3	3	3.9
Confidence Interval	0.0; 1.1	0.0; 2.1	0.0; 3.6	0.0; 5.0	0.0; 6.0

Table 13: RMSE values with confidence intervals for predicting the values of a researcher's h-index in each of the 5 years after their first 10 years.

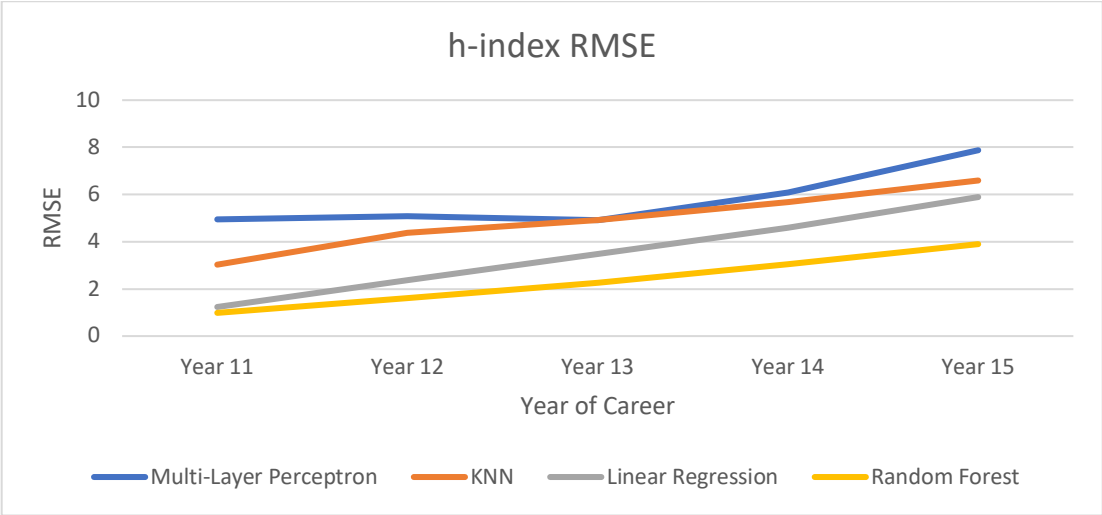

Figure 22: RMSE values for predicting the values of a researcher's h-index in each of the 5 years after their first 10 years.

Table 14 and Figure 23 show the RMSE values for predicting the number of documents a researcher will publish between the 10th and 15th years of their career. The performance of the K-Nearest Neighbours algorithm is improved relative to the other algorithms compared to the same predictions for between the 5th and 10th year where it was a clear outlier. In this set of predictions, it is the Linear Regression that performs poorly, with an RMSE of 15.1 in year 15 while the next worse performing algorithm, the Multi-Layer Perceptron, has a score of 8.7, a 42% improvement. The Multi-Layer Perceptron, Random Forest and K-Nearest Neighbours algorithms all perform differently in different years, indicating that no one algorithm is better suited than others for this task. The best RMSE in year 15 of 6.7 is larger than the average number of publications in the same year, 5.6, but it is worth noting that this is less than the standard deviation for publications in the 15th year, 7.6.

Publication Count RMSE	**Year 11**	**Year 12**	**Year 13**	**Year 14**	**Year 15**
Multi-Layer Perceptron	4.2	5.3	4.2	5.2	8.6
Confidence Interval	1.0; 6.0	0.0; 9.1	1.0; 6.1	1.0; 6.1	0.0; 11.1
K-Nearest Neighbours	6.3	5.0	5.9	6.2	7.1
Confidence Interval	0.0; 8.1	0.0; 8.0	0.0; 7.1	1.0; 11.0	0.0; 9.0
Linear Regression	3.8	8.5	12.9	13.5	15.1
Confidence Interval	0.0; 5.1	0.0; 16.2	0.0; 24.2	0.0; 25.3	0.0; 28.4
Random Forest	3.5	4.6	5.5	5.0	6.7
Confidence Interval	0.0; 5.0	0.0; 7.1	0.0; 8.2	0.0; 7.0	0.0; 8.1

Table 14: RMSE values with confidence intervals for predicting the values of a researcher's publication count in each of the 5 years after their first 10 years.

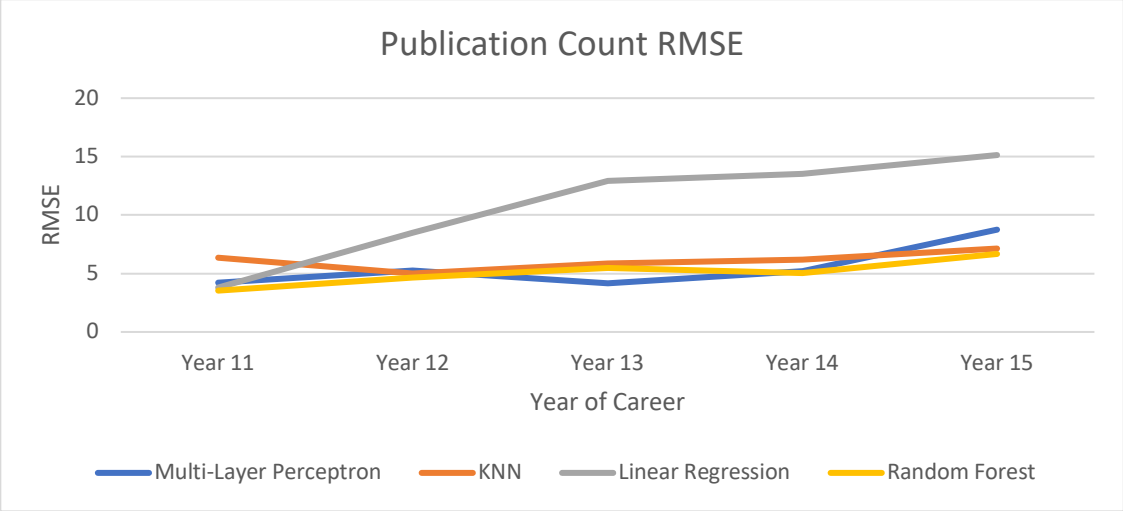

Figure 23: RMSE values predicting the values of a researcher's publication count in each of the 5 years after their first 10 years.

Table 15, Table 16, Figure 24 and Figure 25 show the MAPE and RMSE values of predicting the number of citations a researcher will attain between the 10th and 15th years of their career. The graphs show the same trend as was seen in the same predictions for between years 5 and 10, albeit less extremely. The best algorithm as measured by MAPE, K-Nearest Neighbours, is the worst algorithm as measured by RMSE, and the worst as measured by MAPE, Linear Regression, performs similarly to the Multi-Layer Perceptron and Random Forest algorithms as measured by RMSE.

These results tell mostly the same story as their counterparts for predicting between 5 and 10 years. Reusing predictions as training values will only serve to compound errors. The best prediction for citation count as measured by RMSE, the Multi-Layer Perceptron, has a score of

Citation Count MAPE	**Year 11**	**Year 12**	**Year 13**	**Year 14**	**Year 15**
Multi-Layer Perceptron	70	124	105	160	214
Confidence Interval	6; 125	5; 274	4; 254	8; 285	11; 285
K-Nearest Neighbours	49	76	82	90	117
Confidence Interval	2; 107	2; 160	4; 188	7; 224	15; 242
Linear Regression	58	130	200	331	353
Confidence Interval	5; 100	4; 254	4; 387	6; 540	7.337; 668
Random Forest	50	82	97	143	126
Confidence Interval	3; 134	4; 150.	11; 153	10; 323	15; 279

Table 15: MAPE values with confidence intervals for predicting the values of a researcher's citation count in each of the 5 years after their first 10 years.

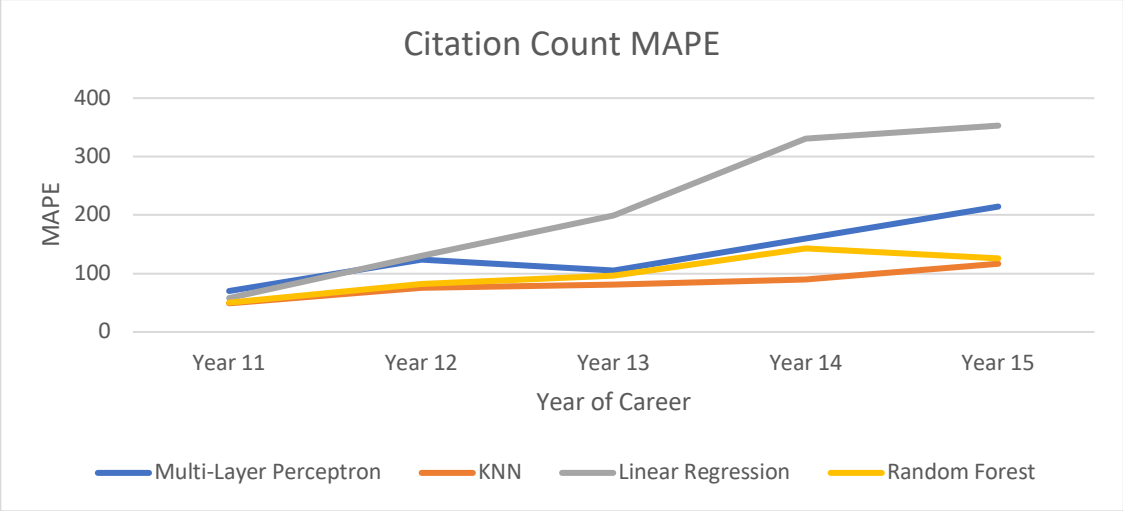

Figure 24: MAPE values for predicting the values of a researcher's citation count in each of the 5 years after their first 10 years.

Citation Count RMSE	**Year 11**	**Year 12**	**Year 13**	**Year 14**	**Year 15**
Multi-Layer Perceptron	72	112	180	225	258
Confidence Interval	2; 89	3; 128	3; 183	8; 250	14; 331
K-Nearest Neighbours	78	132	229	296	386
Confidence Interval	1; 101	1; 191	2; 282	2; 442	5; 624
Linear Regression	59	95	148	252.	352
Confidence Interval	3; 89	3; 104	5; 190	11; 310	17; 433
Random Forest	65	107	167	252	331
Confidence Interval	2; 93	3; 129	4; 235	8; 336	11; 413

Table 16: RMSE values with confidence intervals for predicting the values of a researcher's citation count in each of the 5 years after their first 10 years.

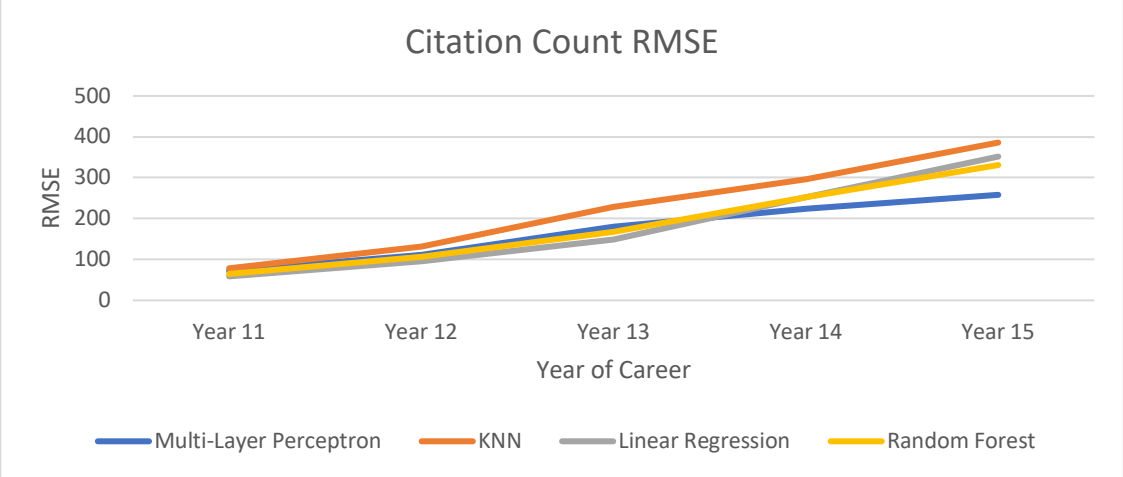

Figure 25: RMSE values with confidence intervals for predicting the values of a researcher's citation count in each of the 5 years after their first 10 years.

258, 40% of the best RMSE for predicting the total citation count in year 15, 652. These results would indicate that 5 separate predictions based on the first 10 years of data would provide better results than including predicted values in the training data. Even without comparing to the other predictions, the RMSE of 258 achieved by the Multi-Layer Perceptron is only an average score when the average citation count in year 15 is 369 and the standard deviation is 593.

5.3.3 15 to 20 Years

Table 17 and Figure 26 show the RMSE scores attained by each algorithm predicting the h-index of a researcher for each year between the 15th and 20th years of their career. The same trends as were seen in the same results for the previous two time periods are also seen here, with Linear Regression and Random Forest performing the best of the four algorithms. Both have very similar scores and confidence intervals. Linear Regression's RMSE of 2.9 in year 20 is only 0.3 less than Random Forest's score of 3.2, and there is only 0.1 between the lower bound of their confidence intervals. These scores are particularly good given the mean of 18.7 and standard deviation of 12.9 for researchers' h-indexes in their 20th year.

h-index RMSE	**Year 16**	**Year 17**	**Year 18**	**Year 19**	**Year 20**
Multi-Layer Perceptron	1.7	5.7	10.0	9.4	14.3
Confidence Interval	0.0; 3	0.6; 11.1	2.6; 13.4	2.0; 15.0	1.6; 17.8
K-Nearest Neighbours	3.2	4.0	4.8	5.7	6.9
Confidence Interval	0.0; 4.0	0.0; 5.5	0.0; 6.7	0.6; 9	0.0; 10
Linear Regression	0.94	1.355	1.746	2.174	2.9
Confidence Interval	0.0; 1.5	0.0; 2	0.0; 3	0.0; 4	0.0; 4.5
Random Forest	1.0	1.4	2.1	2.5	3.2
Confidence Interval	0.0; 1.5	0.0; 2	0.0; 3	0.0; 4	0.1; 4.5

Table 17: RMSE values with confidence intervals for predicting the values of a researcher's h-index in each of the 5 years after their first 15 years.

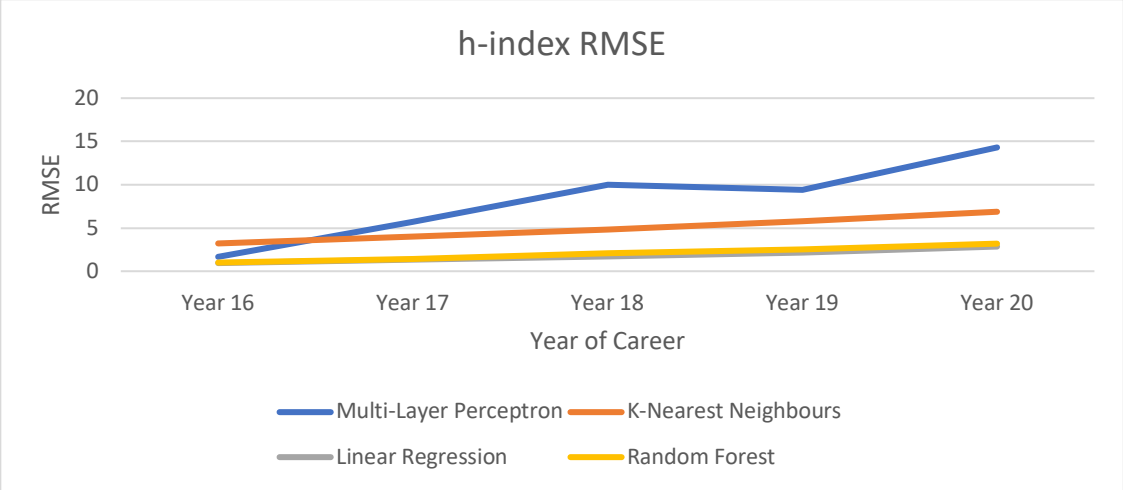

Figure 26: RMSE values for predicting the values of a researcher's h-index in each of the 5 years after their first 15 years.

Table 18 and Figure 27 show the RMSE scores attained by each algorithm predicting the publication count of a researcher for each year between the 15th and 20th years of their career. Compared to their performances on the 10 to 15 year time period, the Multi-Layer Perceptron and Linear Regression algorithms have swapped place, with Linear Regression performing the best with a score of 4.7 in year 20 and the Multi-Layer Perceptron performing the worst with a score of 13.4. This fits with the theory that a linear model is more suited to predicting values for late in a researcher's career than earlier, as a researcher becomes less likely to see a sudden increase or decrease in popularity or productivity. Linear Regression's RMSE of 4.7 in year 20, when the average researcher publishes 6.6 documents with a standard deviation of 7.3, makes it a somewhat reliable algorithm.

Publication Count RMSE	**Year 16**	**Year 17**	**Year 18**	**Year 19**	**Year 20**
Multi-Layer Perceptron	9.2	8.0	13.1	13.9	13.4
Confidence Interval	0; 16	0; 8.5	0; 18	0; 19	0; 17.5
K-Nearest Neighbours	5.8	5.5	5.8	7.2	9.2
Confidence Interval	0; 10	0.5; 8	0; 7.5	1; 10	0; 10
Linear Regression	3.4	3.9	4.8	4.1	4.7
Confidence Interval	0; 5	0; 5.5	0; 7.5	1; 6.5	0; 7
Random Forest	3.8	4.7	6.7	5.6	6.8
Confidence Interval	0; 7	0; 8.5	0.5; 11	0; 9	0.5; 8.5

Table 18: RMSE values with confidence intervals for predicting the values of a researcher's publication count in each of the 5 years after their first 15 years.

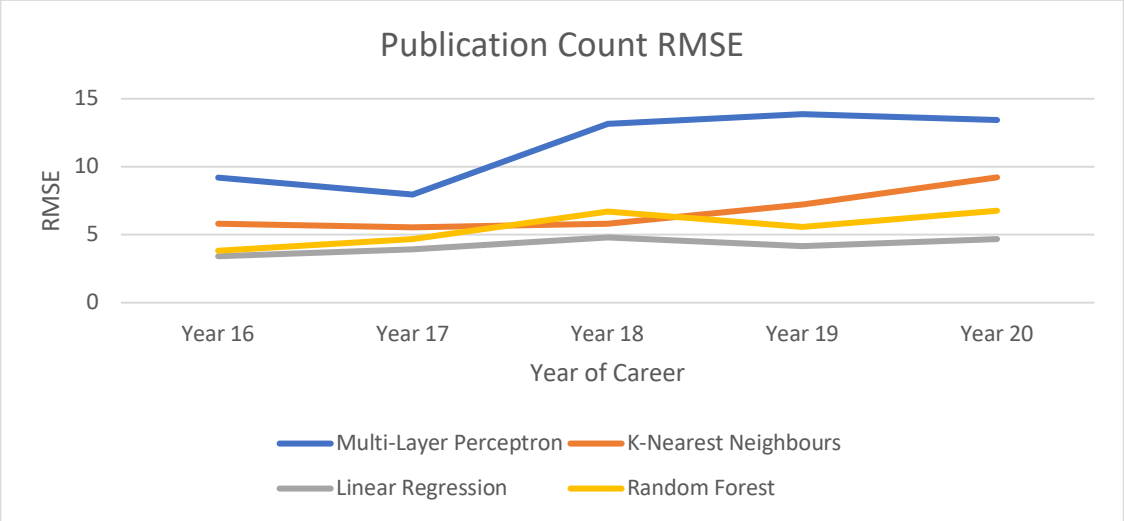

Figure 27: RMSE values for predicting the values of a researcher's publication count in each of the 5 years after their first 15 years.

Table 19, Table 20, Figure 28 and Figure 29 show the MAPE and RMSE values for predicting a researchers citation count in each year between the 15th and 20th year of their career. The pattern of performance measured by MAPE reversing when measured by RMSE, as seen in previous time periods, is not continued here. Random Forest is the worst performing algorithm as measured by both metrics, while the other three perform similarly. The difference between the best MAPE score in year 20, K-Nearest Neighbours with 86 and the second worst, the Multi-Layer Perceptron with 117 is only 31, while the difference between the best RMSE score in year 20, the Multi-Layer Perceptron with a score of 330 and the third worst, 390, is only 60. These close scores coupled with the wide confidence intervals mean that it is difficult to definitively decide which algorithm performs best overall. The mean value for citations received in a researcher's 20th year is 486, with a standard deviation of 760.4. Even the best performing RMSE values are near this mean, indicating that the results are not accurate, which is also supported by the high MAPE values. This further reinforces the findings from the previous two time periods analysed, that compound predictions are less accurate than single predictions.

Citation Count MAPE	Year 16	Year 17	Year 18	Year 19	Year 20
Multi-Layer Perceptron	38	73	107	79	117
Confidence Interval	3; 90	3; 169	7; 319	7; 156	7; 279
K-Nearest Neighbours	40	50	58	68	86
Confidence Interval	3; 58	5; 88	3; 100	9; 145	8; 197
Linear Regression	33	47	58	83	102
Confidence Interval	2; 87	3; 100	2; 105	6; 210	7; 214
Random Forest	68	81	85	154	222
Confidence Interval	5; 132	5; 128	3; 167	7; 495	12; 544

Table 19: MAPE values with confidence intervals for predicting the values of a researcher's citation count in each of the 5 years after their first 15 years.

Citation Count RMSE	Year 16	Year 17	Year 18	Year 19	Year 20
Multi-Layer Perceptron	71	114	1777	244	330
Confidence Interval	3; 93	4; 156	8; 213	9; 302	11; 404
K-Nearest Neighbours	75	129	221	298	387
Confidence Interval	2; 124	3; 178	2; 229	3; 301	7; 528
Linear Regression	99	155	229	323	390
Confidence Interval	2; 137	3; 194	4; 334	5; 569	12; 694
Random Forest	447	510	520	540	604
Confidence Interval	2; 811	2; 940	1; 970	6; 1020	12; 1149

Table 20: RMSE values with confidence intervals for predicting the values of a researcher's citation count in each of the 5 years after their first 15 years.

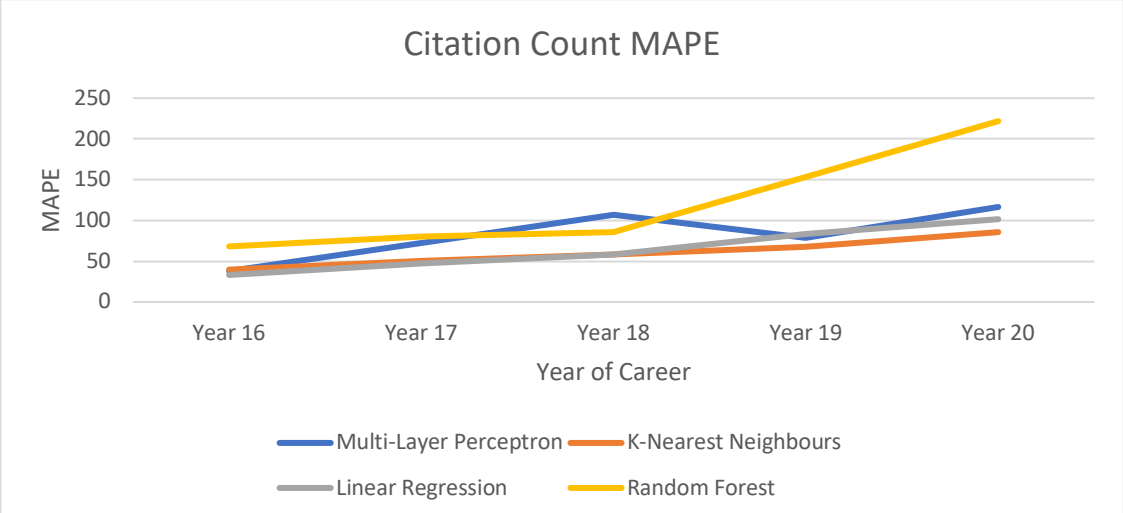

Figure 28: MAPE values for predicting the values of a researcher's citation count in each of the 5 years after their first 15 years.

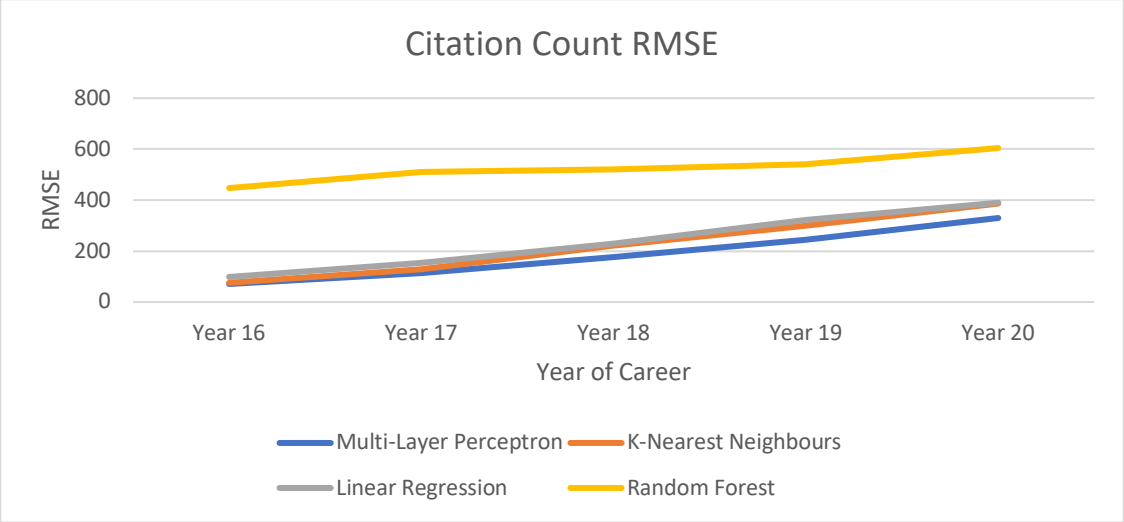

Figure 29: RMSE values for predicting the values of a researcher's citation count in each of the 5 years after their first 15 years.

5.4 Employment Prediction

The following sections show the performance of different K-Nearest Neighbours style algorithms on the employment dataset, in an attempt to predict where a researcher might work in the future. The algorithms are described in Section 3.4.2.

5.4.1 Baseline Predictions

Table 21 and Figure 30 show the performance of the baseline algorithms. Using the Jaccard similarity between different instances as a weighting scheme provided much improved results over a uniform weighting scheme, increasing the number of predictions containing at least one future university to 19.5% from 12%. The use of TF-IDF universally decreased performance, decreasing to 16.1% percent from 19.5% when used with the Jaccard weight and to 11% from 12% using the uniform weighting. Of the four variants tested here, The Jaccard weighting scheme is the best performing variant.

Weighting Scheme	**Jaccard Weight**	**Uniform Weight**	**TF-IDF(Jaccard Weight)**	**TF-IDF(Uniform Weight**
Percentage	19.5%	12.0%	16.1%	11.0%

Table 21: Percentage of predictions with at least one future university, using the Jaccard similarity between researchers as the distance metric

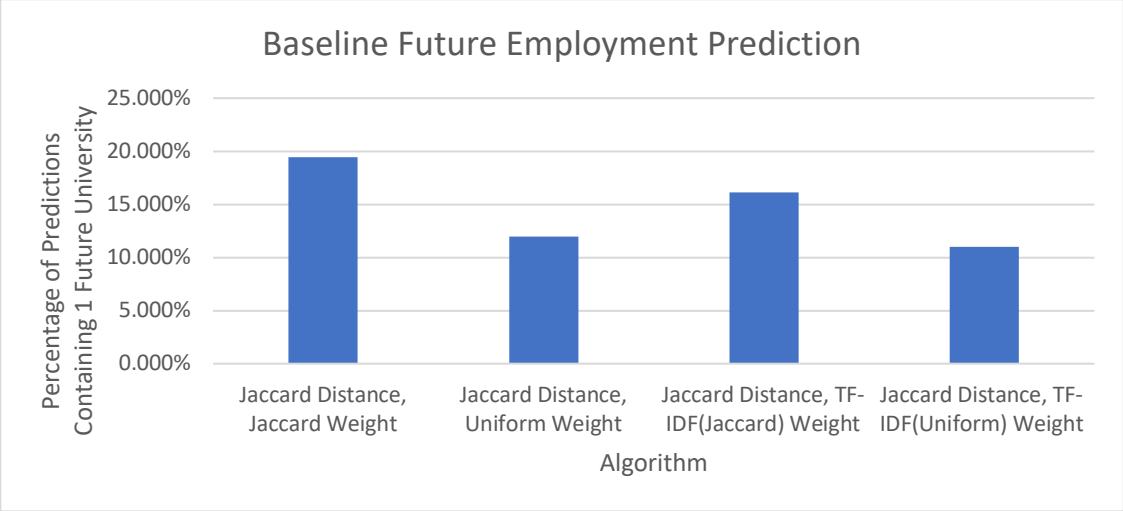

Figure 30: Percentage of predictions with at least one future university, using the Jaccard similarity between researchers as the distance metric

5.4.2 Predictions Using Order

More sophisticated algorithms that consider the order of the universities a researcher has attended were developed to improve these results, as described in Section 3.4.2. The results of the first set of these algorithms, which make use of both the strict and loose order distance metric and use the first 2 universities a researcher attended as training data are displayed in Table 22 and Figure 31. This table and graph compare the percentage of predictions from these algorithms which contain at least one university that a researcher will attend with the best algorithm from the baseline predictions. It is clear from the graph and the table that making use of the order provides a huge boost to the predictive power. The highest percentage, achieved by using the loose order distance metric and a uniform weighting scheme, is 60.3%, 40.8 percentage points higher than the best performing algorithm from the baseline algorithms. Interestingly, the new algorithms see a decrease in performance when distance weighting is used, the opposite to the effect seen on the baseline algorithms. The loose order algorithm drops from 60.3% to 56.3% when a distance weighting scheme replaces the uniform weighting.

Algorithm Variant	**Percentage**
Unordered, Jaccard Distance, Jaccard Weight	19.5%
Strict Order, Split at 2, Uniform Weight	55.5%
Loose Order, Split at 2, Uniform Weight	60.3%
Strict Order, Split at 2, Distance Weight	43.1%
Loose Order, Split at 2, Distance Weight	56.3%

Table 22: Percentage of predictions with at least one future university for different weighting and ordering schemes

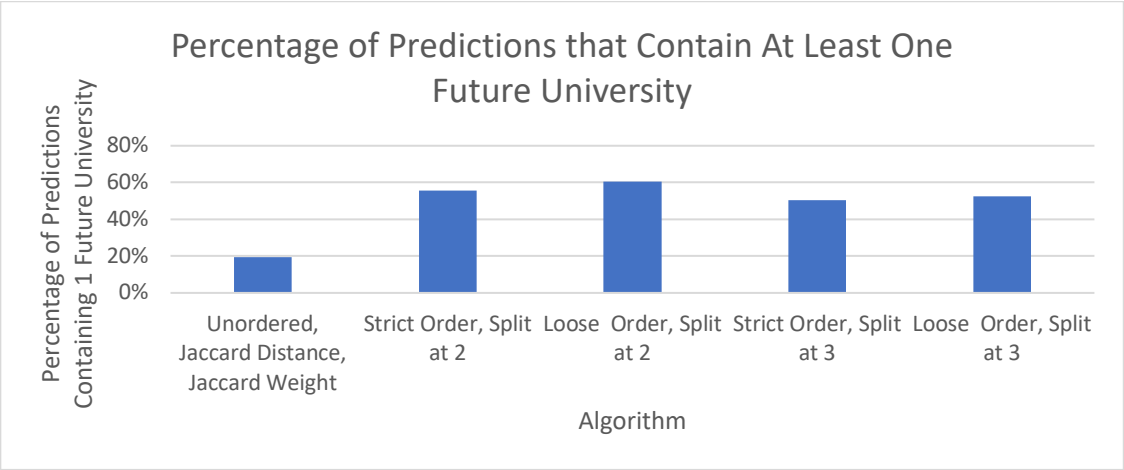

Figure 31: Percentage of predictions with at least one future university for different weighting and ordering schemes

While these results show that uniform weighting produces superior results to distance weighting, the amount of universities to use as training data is another variable that may affect performance. The algorithms were run again using the first 3 universities attended by a researcher as training data and assessed under two new metrics as well as the same one used previously. The results are shown in Table 23 and Figure 32. Increasing the number of training universities worsens performance as measured by every metric. This is not usually the case with machine learning algorithms, where more training data often leads to better results, but the nature of these predictions means that the more universities are used for training, the less are available to appear in the list of 10 possible future universities.

The best algorithm of these results is still the loose order variant, using two universities as training data and a uniform weighting scheme. 60.3% of predictions made by this algorithm contain at least one university where the researcher would go on to work. The average prediction of 10 universities contains 1 future university, although there is a wide confidence interval on this value, between 0 and 3. Each set of 10 universities contained an average of 15.1% of the universities a researcher would go on to work at, again with a wide confidence interval of between 0% and 40%.

Algorithm Variant	**Strict Order Split at 2**	**Loose Order Split at 2**	**Strict Order Split at 3**	**Loose Order Split at 3**
Percentage of Predictions that Contain At Least One Future University	55.5%	60.3%	50.8%	52.6%
Percentage of Future Universities Contained in Each Prediction	12.9% 0%; 33.33%	15.1% 0%; 40%	12.9% 0%; 40%	14.0% 0%; 40%
Number of Future Universities Contained in Each Prediction	0.9 0; 2	1.1 0; 3	0.8 0; 2	0.9 0; 2

Table 23: Scores for different distance metrics and split points with confidence intervals

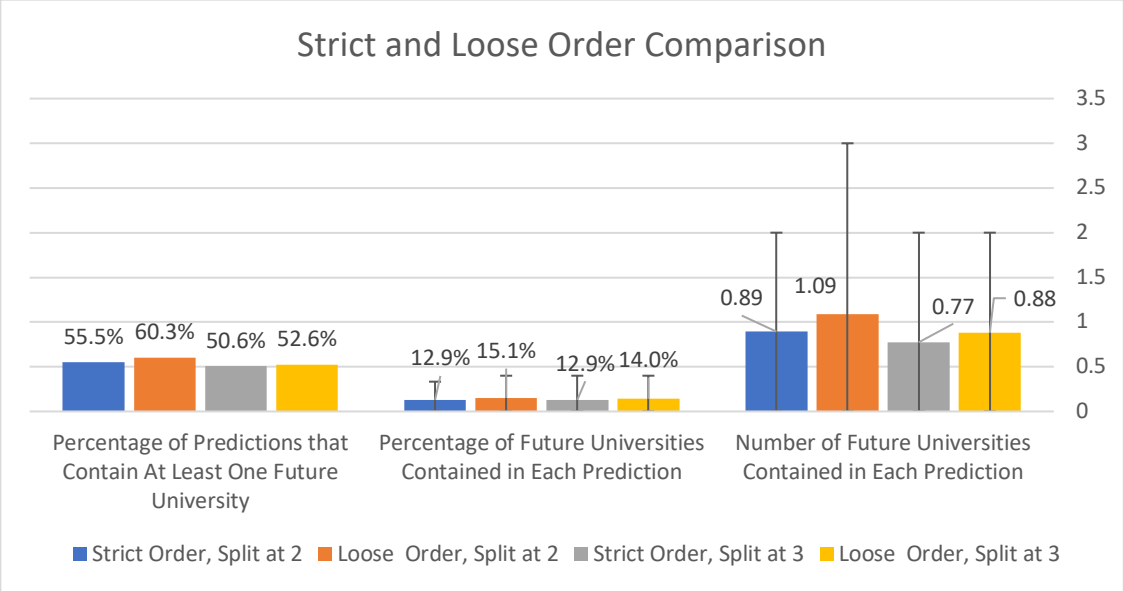

Figure 32: Scores for different distance metrics and split points with confidence intervals

6 Limitations and Future Work

The biggest limitation on these predictions is the amount of data that was available to be analysed. The request limit imposed by the Scopus API limited the number of researchers that could be analysed for citation metrics, and many of the features identified in the ideal model were not available at all. Future work would then first and foremost focus on running these algorithms again on larger datasets with more features, to see how the small dataset size affects performance. The first step in this is to combine the metric and employment datasets, using the rankings of universities a researcher has worked at to predict their citation count, publication count and h-index, while also using these metrics to produce universities they may go on to work at in the future. After this, more algorithms could be compared with these new results.

It's possible that while ARIMA models and Recurrent Neural Networks were unable to be fit to citation histories in this project, other time series algorithms may be found that can work with them. If the same predictions outlined in these results can be improved, then the next step is to expand on the number of features predicted. Publication count, h-index and any new features that are incorporated as training data can be predicted at 5 year intervals similar to how citation count is in this project.

Aside from the possible uses detailed in Section 1 of this dissertation, more advanced versions of this method could be used to make a wide range of predictions. Improvements could be made to existing research paper recommender systems [30-33]. This would allow papers to be recommended based on the projected career of the authors, amplifying the publications of authors who have not had a long publication career but have a high potential. The ability to predict future career metrics could also be used proactively rather than reactively. A system could be created where predictions are made on many researchers at once, allowing universities to identify potential candidates for different positions. Similarly, making predictions for the researchers already working in a university would allow the identification of promising researchers who could be potentially awarded more funding or resources before they have proven themselves traditionally.

One thing that is important to bear in mind regarding any future applications, like many predictive systems, is that their predictions are not the only possible version of events. Any implementation of systems as described above would have to be done with the understanding that researcher's predicted to have good careers may not, and that those predicted to have poor careers may become successful. It will always be important to have a human analyse the predictions made when they can affect important aspects of a person's life like their job or in the case of academic researchers, the funding or resources allocated to them.

7 Conclusions

The stated research goal for this project was to "investigate the ability of machine learning and time series techniques to predict academic careers, with the ultimate aim of creating a method that can reliably predict how certain metrics of an academic career will change as time passes.". This dissertation has thoroughly investigated multiple algorithms and implementations that certainly make steps towards this goal, although none of them can be described as a complete model of academic career prediction.

The most extensive work was done in predicting a researcher's citation count 5 years after the 5th, 10th and 15th years of their publishing career. The results of this work are shown in Figure 33 and Table 24 below.

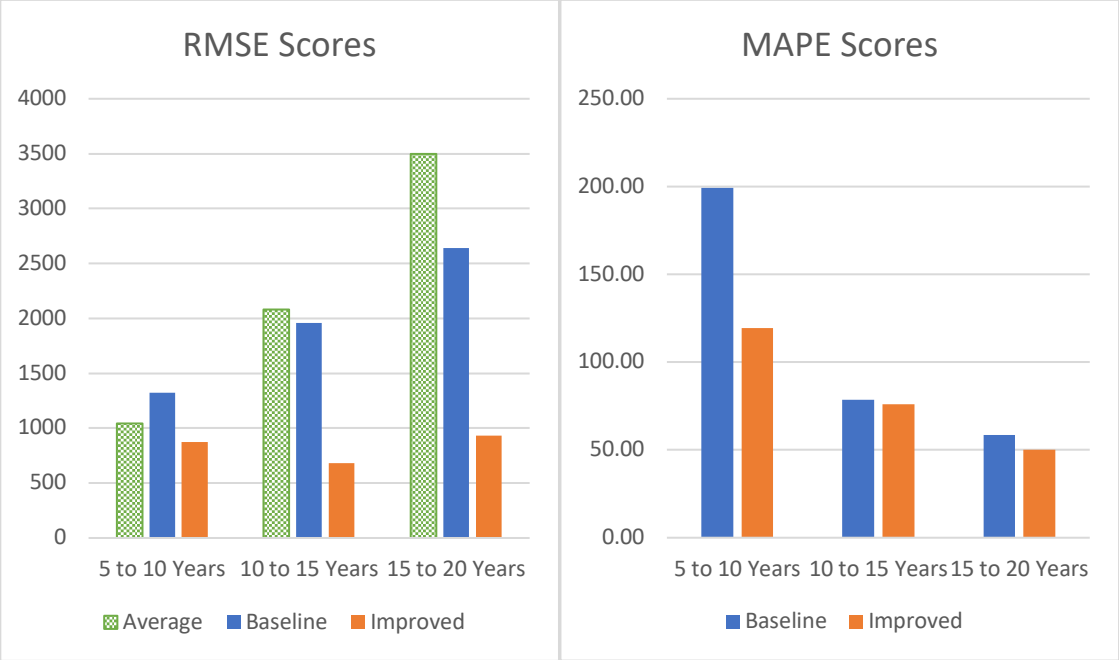

Figure 33: Comparison of RMSE and MAPE scores of best performing algorithms for different time periods

	5 to 10 Years	**10 to 15 Years**	**15 To 20 Years**
Best Algorithm	Multi-Layer Perceptron	Multi-Layer Perceptron	Linear Regression
Lag size	1 Year Lag	7 Year Lag	3 Year Lag
Dataset Variant	Individual Years	Individual Years - Difference	Individual Years - Difference
MAPE	119	76	50
Confidence Interval	52; 56	40; 128	30; 66
RMSE	876	682	934
Confidence Interval	287; 429	516; 811	487; 1108

Table 24: Comparison of RMSE and MAPE scores of best performing algorithms for different time periods

Figure 33 shows a clear decreasing trend in MAPE as a researcher's career progresses. The small lag sizes used to produce the optimal results imply that it is not the additional training years that improve predictions, but in fact that the opposite holds: less recent years introduce noise into the dataset which can interfere with results. One possible explanation for this downward trend is that a researcher's career becomes more predictable as they continue to publish. The longer their career goes on, the less likely they are to receive a sudden or unexpected surge or drop in citations per year. The RMSE figures are more difficult to interpret

but when compared to the mean and standard deviation of the variables being predicted the same trend of increasing predictability with time becomes apparent again. Another trend shown in Table 24 is the good performance of the individual years and the difference datasets. It is worth noting that the second-best algorithms for each time period, not shown in the previous sections, also ran one of these dataset variants, showing that encoding data with the cumulative values for each metric does not improve performance.

As well as predicting citation counts between these intervals, work was done to predict the evolution of individual metrics in each of 5 successive years, using each predicted value as training data to predict the next years values. Aside from the h-index, which was predicted reasonably accurately, attempts to predict other metrics were average at best. The compounding errors for each year were too much to overcome, leading to large errors by the 5th year. The ability to accurately predict h-index is especially remarkable given that the values were predicted based on the predicted values of citation and publication count, which had very high RMSE and MAPE scores.

The final set of predictions aimed to predict which universities a researcher would work at based on the universities they had worked at previously. The K-Nearest Neighbours Style model created returns a list of 10 possible future universities, with 60% of predicted lists from the best algorithm containing at least one university that the researcher would go on to work at. This algorithm was a huge improvement over the baseline algorithms which did not consider the order in which the universities were worked at, indicating that this order is an important indicator of a researcher's career trajectory.

Of the three types of predictions, these future employment predictions were by far the most successful. While the ultimate research goal of creating a method to predict a complete model of an academic career was not fulfilled, there are solid foundations here to build on in the future. Citation count, h-index and publication count have all been confirmed as features that improve predictions compared to predicting based on only one variable. Compound predictions were found to be far less accurate than single predictions over the same time period. It was demonstrated that recent years are the most important when attempting to predict future citation count and that years from far in the past can actually act as noise. Later stages of a researcher's career were found to be more predictable than early stages. Despite being unable to fit citation counts into a stationary model, the common time series analysis technique of differencing was applied to predicting citation counts and improved performance. In terms of predicting future employment, the order of previous universities attended was shown to make a huge impact on a researcher's future career. It is unfortunate that the citation results were not better, largely due to limitations placed on the size of the dataset and the features available for

training, but the results of these experiments show that predicting these values is indeed possible.

8 Appendix I

The following tables show the results of the best performing time lag for each algorithm on each dataset variant for each of the three time periods examined for the single predictions.

5 to 10 Individual	**MAPE**	**RMSE**
Linear Regression	212	750
2 Year Lag	93; 449	372; 1233
K-Nearest Neighbours	140	828
5 Year Lag	75; 220	434; 1129
Random Forest	211	937
2 Year Lag	116; 296	428; 1830
Multi-Layer Perceptron	140	788
1 Year Lag	79; 199	410; 1507

5 to 10 Cumulative	**MAPE**	**RMSE**
Linear Regression	215	784
3 Year Lag	131; 338	448; 1022
K-Nearest Neighbours	158	873
4 Year Lag	92; 234	392; 1141
Random Forest	180	1047
4 Year Lag	85; 258	600; 1817
Multi-Layer Perceptron	166	769
2 Year Lag	93; 238	439; 1322

5 to 10 Individual Difference	**MAPE**	**RMSE**
Linear Regression	199	715
2 Year Lag	138; 275	326; 1011
K-Nearest Neighbours	131	819
3 Year Lag	73; 190	390; 1162
Random Forest	186	1016
3 Year Lag	76; 337	432; 2329
Multi-Layer Perceptron	119	876
1 Year Lag	67; 175	589; 1304

5 to 10 Cumulative Difference	**MAPE**	**RMSE**
Linear Regression	297	699
3 Year Lag	144; 568	423; 925
K-Nearest Neighbours	193	822
4 Year Lag	101; 302	381; 1216
Random Forest	241	902
4 Year Lag	98; 384	474; 1325
Multi-Layer Perceptron	202	722
4 Year Lag	130; 298	297; 948

10 to 15 Individual	**MAPE**	**RMSE**
Linear Regression	137	71
6 Year Lag	49; 213	595; 844
K-Nearest Neighbours	57	903
5 Year Lag	37; 79	627; 1161
Random Forest	68	935
7 Year Lag	44; 85	502; 1225
Multi-Layer Perceptron	84	683
6 Year Lag	51; 123	448; 861

10 to 15 Cumulative	**MAPE**	**RMSE**
Linear Regression	134	706
6 Year Lag	74; 196	553; 838
K-Nearest Neighbours	75	1176
2 Year Lag	52; 113	633; 1901
Random Forest	75	869
5 Year Lag	49; 101	458; 1167
Multi-Layer Perceptron	80	827
4 Year Lag	45; 124	549; 1305

10 to 15 Individual Difference	**MAPE**	**RMSE**
Linear Regression	127	707
6 Year Lag	48; 235	568; 901
K-Nearest Neighbours	61	889
8 Year Lag	46; 80	536; 1338
Random Forest	66	958
2 Year Lag	43; 106	523; 1397
Multi-Layer Perceptron	76	682
7 Year Lag	39; 128	516; 811

10 to 15 Cumulative Difference	**MAPE**	**RMSE**
Linear Regression	137	73
7 Year Lag	64; 223	45; 116
K-Nearest Neighbours	73	1079
3 Year Lag	45; 116	614; 1716
Random Forest	83	1168
7 Year Lag	42; 142	752; 1398
Multi-Layer Perceptron	77	734
3 Year Lag	53; 122	550; 964

15 to 20 Individual	**MAPE**	**RMSE**
Linear Regression	66	1016
7 Year Lag	41; 103	660; 1346
K-Nearest Neighbours	41	1785
2 Year Lag	32; 53	645; 2481
Random Forest	46	1616
2 Year Lag	36; 60	690; 2312
Multi-Layer Perceptron	39	1012
2 Year Lag	21; 657	793; 1450

15 to 20 Cumulative	**MAPE**	**RMSE**
Linear Regression	61	1090
3 Year Lag	34; 76	745; 1340
K-Nearest Neighbours	68	2061
3 Year Lag	34; 151	626; 3641
Random Forest	69	2237
2 Year Lag	55; 85	1108; 2629
Multi-Layer Perceptron	39	1512
1 Year Lag	28; 46	718; 2269

15 to 20 Individual Difference	**MAPE**	**RMSE**
Linear Regression	50	934
3 Year Lag	30; 66	487; 1108
K-Nearest Neighbours	46	1489
6 Year Lag	33; 58	850; 2400
Random Forest	47	1451
7 Year Lag	26; 80	737; 1862
Multi-Layer Perceptron	35	1072
1 Year Lag	21; 65	702; 1384

9 References

1. Abbot, A., et al., *Do Metrics Matter?* Nature, 2010. **465**(17): p. 3.
2. Morgan, C., *Athletes and Oracles: The transformation of Olympia and Delphi in the Eighth Century BC*. 1990. p. 148.
3. Kononenko, I., *Machine Learning for Medical Diagnosis: History, State of the Art and Perspective.* Artificial Intelligence in Medicine, 2001. **23**(1): p. 11.
4. Patel, J., et al., *Predicting Stock Market Index Using Fusion of Machine Learning Techniques.* Expert Systems with Applications, 2015. **42**(4): p. 11.
5. Beel, J., et al., *Research Paper Recommender Systems: A Literature Survey.* International Journal on Digital Libraries, 2015. **17**(4): p. 33.
6. van Dijk, D., O. Manor, and L.B. Carey, *Publication Metrics and Success on the Academic Job Market.* Current Biology, 2014. **24**(11): p. 6.
7. Daud, A., et al., *Using Machine Learning Techniques for Rising Star Prediction in Co-Author Network.* Scientometrics, 2014. **102**(2): p. 25.
8. Fu, L.D. and C.F. Aliferis, *Using content-based and bibliometric features for machine learning models to predict citation counts in the biomedical literature.* Scientometrics, 2010. **85**(1): p. 13.
9. Tkaczyk, D., B. Tarnawski, and L. Bolikowski, *Structured affiliations extraction from scientific literature.* D-Lib Magazine, 2015. **21**: p. 2.
10. Hirsch, J.E., *An Index to Quantify an Individual's Scientific Research Output.* Proceedings of the National Academy of Sciences of the United States of America, 2005. **102**(46): p. 4.

11. Hirsch, J.E., *Does the h index have predictive power?* Proceedings of the National Academy of Sciences, 2007. **104**(49): p. 6.
12. Acuna, D.E., S. Allesina, and K.P. Kording, *Predicting Scientific Success.* Nature, 2012. **489**: p. 2.
13. Bertsimas, D., et al., *Tenure Analytics: Models for Predicting Research Impact.* Operations Research, 2014. **63**(6): p. 16.
14. Mazloumian, A., *Predicting Scholars' Scientific Impact.* PLoS ONE, 2012.
15. Laurance, W.F., et al., *Predicting Publication Success for Biologists.* BioScience, 2013. **63**(10): p. 7.
16. McCleary, R., et al., *Applied Time Series Analysis for the Social Sciences*. 1980: Sage Publications. 325.
17. Hyndman, R.J. and G. Athanasopoulos, *Forecasting: Principles and Practice*. 2012.
18. Raman, H. and N. Sunilkumar, *Multivariate Modelling of Water Resources Time Series Using Artificial Neural Networks.* Hydrological Sciences Journal, 1994. **40**(2): p. 9.
19. Chakraborty , K., et al., *Forecasting the Behaviour of Multivariate Time series using Neural Networks.* Neural Networks, 1992. **5**(6): p. 10.
20. Sinn, M., et al., *Predicting arrival times of buses using real-time GPS measurements*, in *15th International IEEE Conference on Intelligent Transportation Systems*. 2012: Anchorage, Alaska, USA.
21. Harzing, A.W. *Publish or Perish*. [cited 2018 March 23rd].
22. Tkaczyk, D., et al., *Machine Learning vs. Rules and Out-of-the-Box vs. Retrained: An Evaluation of Open-Source Bibliographic Reference and Citation Parsers*, in *Proceedings of Joint Conference on Digital Libraries*. 2018: Fort Worth, Texas.
23. Lipinski, M., et al., *Evaluation of Header Metadata Extraction Approaches and Tools for Scientific PDF Documents*, in *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*. 2013: Indianapolis, Indiana. p. 2.
24. Beel, J., et al., *Docears PDF Inspector: Title Extraction from PDF files.*, in *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*. 2013. p. 2.
25. Beel, J., et al. *SciPlore Xtract: Extracting Titles from Scientific PDF Documents by Analyzing Style Information (Font Size)*. in *Research and Advanced Technology for Digital Libraries*. 2010. Glasgow, United Kingdom.
26. Beel, J. and B. Gipp. *On the Robustness of Google Scholar Against Spam*. in *21st ACM Conference on Hyptertext and Hypermedia* 2010. Toronto, Canada.
27. Pedregosa, F., et al., *Scikit-learn: Machine Learning in Python.* JMLR, 2011. **12**: p. 6.
28. Seabold, J.S. and J. Perktold, *Statsmodels: Econometric and Statistical Modeling with Python*. 2010, Proceedings of the 9th Python in Science Conference.
29. Chollet, F.a.o., *Keras: The Python Deep Learning library*. 2015, keras.io.
30. Beel, J., *A Macro/Micro Recommender System for Recommendation Algorithms [Proposal]*. ResearchGate, 2017.
31. Beel, J., et al. *Mr. DLib: Recommendations-as-a-Service (RaaS) for Academia*. in *ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*. 2017. Toronto, Canada.
32. Beel, J., et al., *RARD: The Related-Article Recommendation Dataset.* D-Lib Magazine, 2017. **23**(7/8): p. 14.
33. Beel, J. and S. Dinesh. *Real-World Recommender Systems for Academia: The Gain and Pain in Developing, Operating, and Researching them*. in *Fifth*

Workshop on Bibliometric-enhanced Information Retrieval (BIR) co-located with the 39th European Conference on Information Retrieval (ECIR 2017). 2017. Aberdeen, UK.