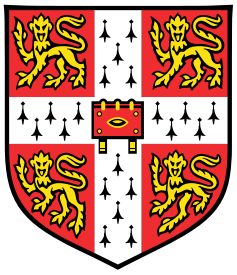
Identifying Valuable Patents: A Deep Learning Approach



Leonidas Aristodemou

Institute for Manufacturing
Department of Engineering
University of Cambridge

This thesis is submitted for the degree of
Doctor of Philosophy

St. Edmund's College September 2020

I would like to dedicate this thesis to my loving wife, Despo, my parents, Paris and Angela, my sister Argyro, and my brother, Marios …

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and acknowledgements. This thesis contains fewer than 65,000 words including appendices, footnotes, tables and equations, but excluding the bibliography, and has fewer than 150 figures.

Leonidas Aristodemou
September 2020

Identifying Valuable Patents: A Deep Learning Approach

Leonidas Aristodemou

Big data is increasingly available in all areas of manufacturing, which presents value for enabling a competitive data-driven economy. Increased data availability presents an opportunity to introduce the next generation of innovative technologies. Firms invest in innovation and patents to increase, maintain and sustain competitive advantage. Consequently, the valuation of patents is a key determinant in economic growth since patents are an important innovation indicator. Given the surge in patenting throughout the world, the interest in the value of patents has grown significantly. Traditionally, studies on patent value have focused on limited data availability restricted to a specific technology area using methods such as regression, and mostly using numeric and binary categoric data types. We propose the definition for *intellectual property intelligence (IPI)* as the data science of analysing large amount of IP information, specifically patent data, with artificial intelligence (AI) methodologies to discover relationships and trends in the data for decision making.

With the rise of AI and the ability to analyse larger datasets of patents, we develop an AI deep learning methodology for the valuation of patents. To do that, we build a large USPTO dataset consisting of all granted patents from 1976-2019: (i) we collect, clean, collate and pre-process all the data from the USPTO (and the OECD patent quality indicators database); (ii) we transform the data into numeric, categoric, and text features so that we are able to input them to the deep learning model. More specifically, we transform the text (abstract, claims, summary, title) into feature vectors using our developed Doc2Vec vector space model (VSM), that we assess using the t-distributed stochastic neighbour embedding (t-SNE) visualisation. The dataset is made publicly available for researchers to efficiently and effectively run fairly complex data analysis.

We propose an AI deep learning methodology for the valuation of patents to identify valuable patents. Using our developed dataset, we build AI deep learning models, which are based on deep and wide feed-forward artificial neural networks (ANN), with dropout, L2 penalty and batch normalisation regularisation layers, to forecast the value of patents with 12 ex-post patent value output proxies. These include the grant_lag, generality, quality_index_4, and forward citations, generality_index and renewals in three time horizons (t4, t8, t12). We associate these patent value proxies to their respective patent value dimension (economic, strategic and technological). We forecast patent value using ex-ante patent value input determinants, for a wide range of technological areas (using the IPC classes), and time horizon domains (short term in t4, medium term in t8, and long term in t12).

We evaluate all our models using a variety of strategies (out-of-time test, out-of-sample test, k-Fold and random split cross validation), and transparently report all metrics (accuracy, confusion matrix, F1-score, false negative rate, log loss, mean absolute error, precision, recall). Our models have higher accuracy and macro average F1-scores, with low values for the training and validation losses compared to prior art. With increasing prediction horizons, we observe an increase in the macro average F1-scores for several of the proxies. In addition, we find that the composite index that takes into consideration more than one value dimension, has the combined highest accuracy and macro average F1-score, relative to single value dimension patent proxies. Moreover, we find that firms seem to file widely at the short term time horizon and then focus their technological competencies to established opportunities. Patent owners seem to renew their patents in the fear of losing out. Our study has moved away from relatively small datasets, limited to specific technology field, and allowed for reproducibility in other fields. We can tailor models to different technology area, with different patent value proxies, with different time horizons.

This study proposes an AI methodology, which is based on deep learning, using deep and wide feed forward artificial neural networks, to predict the value of patents, which has academic and industrial implications. We predict the value of patents with a variety of output proxies, including composite index proxies, for different technology areas (IPC classifications) and time horizons. Since we use all USPTO granted patents from 1976-2019 to train our models, we can apply this approach to patents in any technology field. Our approach enables researchers and industry professionals to value patents using a variety of patent value proxies, based on different value dimensions, tailored to specific technology areas. The proposed AI deep learning approach could effectively support expert decision making (technology, innovation and IP managers etc.) in their decision making by providing fast, low cost, data-driven intellectual property intelligence (IPI) from big patent data. Firms with limited resources, i.e. small-medium enterprises (SMEs) can choose representative proxies to forecast patent value estimates, saving resources. Consequently, the proposed approach could efficiently support experts in their patent value judgement, policy making in the government's investments in technological sectors of the future to support the economy, and patent offices with the AI approaches to analyse efficiently and effectively big patent data. We anticipate this research would be interesting for future researchers to expand the emerging field of IPI research and the skills they will need to perform IPI data-driven research with a variety of data sources and AI deep learning ANN approaches.

Acknowledgements

Ithaka

As you set out for Ithaka
hope your road is a long one,
full of adventure, full of discovery.
...
And if you find her poor, Ithaka won't have fooled you.
Wise as you will have become, so full of experience,
you'll have understood by then what these Ithakas mean.

Constantinos P. Cavafy

Ithaka is one of my favourite poems, because it portrays so nicely the definition of the journey. The journey is equally important as the destination. When I set out four years ago, to complete a PhD in Engineering, my primary aim was to learn new skills, expand my knowledge, and identify opportunities to touch and improve people's lives. The PhD journey has been long and demanding, yet full of experiences and learnings. This thesis and the skills developed have yielded the following: the research code and research dataset (Aristodemou, 2020a,b), 4 journal articles (Aristodemou & Tietze, 2018a,b; Aristodemou et al., 2020; Tietze et al., 2020b), 7 conference articles (Aristodemou & Tietze, 2019a, 2020; Aristodemou et al., 2017a, 2018; Jeong et al., 2019; Li et al., 2020; Silva et al., 2019), 6 working paper articles (Aristodemou & Tietze, 2017a, 2019b; Aristodemou et al., 2017b, 2019a,b; Tietze et al., 2020a), and 1 report (Aristodemou & Tietze, 2017b) until now. With patience, endurance, self-disciple, and family love, I have enjoyed this journey. Most importantly, I am grateful to the people I have met throughout this time, building my professional network and friendships that would last for a lifetime.

First and foremost, I would like to express my sincere gratitude to my supervisor Dr. Frank Tietze. I am extremely thankful for the opportunity he provided me to pursue a PhD, within the Innovation Intellectual Property Management (IIPM), Centre for Technology Management (CTM), Institute for Manufacturing (IfM), Department of Engineering, University of Cambridge. His help, advice and guidance has been invaluable both on a personal

and professional level. I am thankful for his immense knowledge, constant patience and motivation. He has been an incredible source of support and encouragement throughout my PhD studies, and a tremendous mentor for me.

I am also indebted to all my advisers, both within CTM, and across academia and industry. I would like to thank the following for their invaluable help, support, and useful comments and insights: Prof. Tim Minshall, Dr. Rob Phaal, David Probert, Dr. Letizia Mortara, Dr. Chander Velu, Dr. Thomas Bohne, Dr Clive Kerr, Dr. Nicki Athanassopoulou, Dr. Imoh Ilevbare, Prof. Rick Mitchell, Dr. Pratheeba Vimalnath, and Dr. Marina Evangelou. I would also like to thank the people from the STIM consortium and all the conferences I have attended. Furthermore, I would like to thank my two examiners: Prof. Tim Minshall, and Prof. Ove Granstrand for the useful discussion, and fruitful comments in improving my research. Moreover, I would also like to give special thanks to Geraldine Guceri for all her help and support, and all my fellow students at the CTM and the Alan Turing Institute, especially Marc Felkse, Martha Geiger, Philipp Koebnick, Xenia Miscouridou, and Alexis Bellot. I would also like to thank all the faculty of Department of Engineering, and the Institute for Manufacturing, for their helpful comments and advice on improving my research.

I am also grateful to the Engineering and Physical Sciences Research Council (EPSRC) for the DTP studentship award scholarship for fees and maintenance. Additionally, I am thankful to the A. G. Leventis Foundation, the CTM fund, and the Alan Turing Institute enrichment fund. I would also like to thank St. Edmund's college, for providing constant support throughout this time in a friendly and collective environment.

I would also like to thank all my EPONIMOUS friends, and my friends in London, Cambridge, Paris and Cyprus for all the enjoyable and memorable moments we have had.

Most importantly, I am deeply grateful to my loving wife, Despo for her continuous love and support in this journey. She is standing by my side for the rest of time, giving me hope in my times of trial, joy in my saddest hours, and love in all I do. Without her, this thesis would not have been possible. I am also deeply grateful to my parents, Paris and Angela, my sister Argyro and my brother Marios, for their continuous love, encouragement and sacrifices. I am grateful for always believing in me, teaching me to dream with my eyes wide open, and with humility to aim for the sky. Without them, this thesis would not have been possible either. I would also like to thank my uncle Panicos, my aunt Alexia, my cousins Andreas, Amy, Evelyn, Iris, Akis, my grandmother Maro, and my uncle's family, Andreas, Katerina and Marios. I would also like to thank my in-laws, Christos, Afroditi, Onoufrios and Nastazia.

Lastly, I would like to make a special dedication to my grandfather, Leonidas, God rest his soul, who would be extremely proud for what I have achieved, and who taught me to always be 'leventis'. Looking down, I am sure he would be smiling on my graduation day.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Research background

1.1.1 Research positioning

Big data is increasingly available in all areas of manufacturing (OECD, 2017). Data as such presents value for enabling a data-driven economy, at the heart of the Internet of things and Industry 4.0 (EPO, 2016; Gubbi et al., 2013). Increased data availability presents an opportunity for better decision making, policy and strategy development, to introduce the next generation of innovative technologies (Günther et al., 2017; Mowery et al., 1996; Teece, 1986).

In this time of changing technologies, shrinking product lifecycles and growing international competitiveness, it is increasingly important for firms to create and maintain competitive advantage (Grant, 1991). Innovation is a hybrid concept that has evolved over time and adapted itself to changing conditions (Fagerberg et al., 2006). It plays a major role in the growth and economic competitiveness of firms, industries and countries (Grant, 2012). Innovation can also be defined as improvements in technology, regardless of whether the new ideas are embodied in products, processes or services (Chesbrough, 2003a,b). Fagerberg et al. (2006) argues that the function of innovation is to introduce novel knowledge into the economic sphere. The knowledge-based economy is defined as an economy directly based on the distribution and use of knowledge (European Commission, 2004). This knowledge-driven economy is at the heart of the technological era, which strengthens the growth of all economies and sustainability paths (European Commission, 2004).

The increasing importance of knowledge as an economic driver has major implications for innovation management, which is a key determinant of competitiveness. Firms invest in innovation to build knowledge, and increase and sustain competitive advantage. Consequently,

the valuation of innovation, and specifically of technologies, is a key determinant in economic growth (Verhoeven et al., 2016). Several scholars have argued that the value of technologies can be modelled as the value of patented inventions, i.e. patent value, which varies widely at the patent, firm and industry level (Gambardella, 2011; Gambardella et al., 2005; Giuri et al., 2007; Harhoff & Hoisl, 2006). Patents are an important indicator to assess the innovation capabilities of technologies across nations. Many studies use them to map the innovation and technological profile of firms or as indicators for future economic activities (Harhoff et al., 2007). Economic research has looked into the value of patents extensively (Bessen, 2008; Hall, 2005; Harhoff et al., 1999, 2003).

Recently, there has been much discussion about patent value, its definition, how to measure it and what it entails for innovation and technology development (Grimaldi & Cricelli, 2019). Patent value analysis plays an important role in managerial economic and business strategy in that it helps to estimate the value of the technologies. Given the surge in patenting throughout the world (WIPO, 2019a,c, 2020), the interest in the valuation of technologies has grown significantly (Greenhalgh & Rogers, 2006; Lagrost et al., 2010; Pitkethly, 1997). Traditionally, studies on patent value, have focused on limited data availability, and regression analysis (Hall, 2005; Harhoff et al., 1999; Lanjouw & Schankerman, 2004; Reitzig, 2004), with emphasis on market value, and mostly numeric and binary categoric data types (van Zeebroeck, 2011).

Over the last two decades, there have been substantial developments in the field of patent analytics, which describes the science of analysing large amounts of patent data, in relation to other data sources, to discover relationships (Abbas et al., 2014; Baglieri & Cesaroni, 2013; Trippe, 2003). With the digitization of patent data (Dintzner & Van Thieleny, 1991), and gradual improvements of the data quality and analytical techniques over the last decade, the world's largest repository of technical information has become accessible for rapidly decreasing costs and to a wider non-specialist audience (Aristodemou et al., 2017b).

With the rise of Artificial Intelligence (AI)[1], a number of AI methods have been applied to analyse intellectual property data (Abbas et al., 2014; Baruffaldi et al., 2020; Oldham & Fried, 2016; Trappey et al., 2020a; Trippe, 2015). In a recent study, we use the technology roadmapping approach (Phaal et al., 2012) to explore the future of analysing patent data (Aristodemou & Tietze, 2017b; Aristodemou et al., 2017b). We identify 11 priority technologies, such as AI, that industry experts believe to be important to increase their

[1]The term artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to mimic human actions, exhibits traits associated with human minds such as learning and problem solving, and have the ability to rationalise in achieving a specific goal. AI collectively includes machine learning (ML) and deep learning (DL) methods. Bringsjord & Govindarajulu (2020), URL: https://plato.stanford.edu/cgi-bin/encyclopedia/archinfo.cgi?entry=artificial-intelligence.

adoption for the analysis of patent data, since other domains have already adopted widely such technologies (Agatonovic-Kustrin & Beresford, 2000; Alcácer & Cruz-Machado, 2019; Chen et al., 2018; De Fauw et al., 2018; Dernis et al., 2019; OECD, 2019a). In this research, we refine the definition for *Intellectual Property Intelligence (IPI)* as the data science of analysing large amount of intellectual property (IP) information, specifically patent data, with artificial intelligence methodologies such as machine learning and deep learning, to discover relationships, trends and patterns in the data for decision making. This is a subset of the definition proposed by Aristodemou & Tietze (2018b) and in line with the definition of *Patinformatics* (Trippe, 2003).

With the ability to analyse larger datasets of patents, this research brings together the valuation of patents, with AI methodologies (Aristodemou & Tietze, 2018b). Given the limited association and application of AI methodologies to patent value (Choi et al., 2020), we develop a deep learning approach based on a large USPTO dataset consisting of all granted patents from 1976-2019, to forecast patent value, using patent data (the technical information contained within the patent documents). We do that by using deep and wide feed-forward artificial neural networks to forecast the value of patents for a wide range of patent value proxies (and their respective patent value dimension), technological areas (using the IPC classes), and time domains.

1.1.2 Exploring the future of analysing patent data

In a connected world, where successful technological development depends on collaboration of different partners, effectively analysing and valuing patent data has huge, yet only partially exploited potential (Lee et al., 2011). The recent advancements of data technologies, known by the collective term of artificial intelligence (AI) methods, such as machine learning and deep learning, seem to potentially deliver breakthrough progress to enable new use cases for patent data with substantial economic benefits. While these technologies already impact several areas, their impact on patent data remains to be understood in growing the emerging field of AI-based IPI research (Lupu et al., 2011; Trappey et al., 2020b).

We carried an exploratory study to understand the field of analysing patent data (Aristodemou & Tietze, 2017b; Aristodemou et al., 2017b). In the study, we deployed a technology roadmapping approach (Phaal, 2004; Phaal et al., 2001; Probert et al., 2003) through a focus group to produce a technology roadmap for the patent data analytics field. The patent analytics domain technology (PADT) roadmap has a vision of a fully adaptive, interactive, intelligent system, and aims to contribute in helping research and industry to explore potential

breakthroughs. We explored 5 areas: the patent data[1], the data interconnectedness[2], the analysis effectiveness[3], the analysis visualisations[4], and the quality of patents[5] (Abbas et al., 2014; Moehrle et al., 2010; Squicciarini et al., 2013; Trappey et al., 2012).

The PADT roadmap evidently shows that the technology development and integration of artificial intelligence (AI) methods (and specific deep learning and artificial neural networks) can have a significant impact on the development of the field throughout time, with a transparent and consistent reporting of information and a clean standardised interlink patent dataset[6]. This explorative study, with the collective knowledge of 100+ domain experts, and the analysis of the 5 areas above, and the consolidated discussion of the PADT roadmap, has helped to identify the potential of AI methodologies for the analysis of patent data.

[1]The area of patent data focuses on issues arising with data management, data preparation, and data inconsistencies (Baudour & van de Kuilen, 2015; Martinez, 2010; Moehrle et al., 2010). The focus group of experts engaged in discussions to produce a mini-technology roadmap, with a vision of harmonised open source patent data. In terms of technology developments, the experts discussed: (i) the meta-database harmonisation, (ii) the full text analysis with natural language processing and latent semantics methods (including artificial intelligence, machine learning, and artificial neural networks).

[2]The area of database interconnectedness tackles issues with types of data not properly combined. In the focus group, the experts produced a mini-technology roadmap visualising databases that are connected to products and cross-referencing across all data streams, with standardisation, entity disambiguation and technology classification. Natural language processing methods, artificial intelligence and the development of ontologies can contribute on how to best approach this area, where strong cooperation between patent offices is essential to establish and implement common standards.

[3]The area of data analysis effectiveness focuses on understanding and deciding analytical techniques (Abbas et al., 2014), and how to deploy them (Squicciarini et al., 2013). The expert focus group produced a mini-technology roadmap that envisions a fully automated, highly intelligent, highly adaptive artificial intelligence system. The experts discussed how an expanded understanding of the existing analytic tools and techniques can be helpful for both industry and academia (Oldham & Fried, 2016; Trippe, 2015). They identified that technologies such as artificial intelligence, machine learning and deep learning could revolutionise the domain, with open source and open data as facilitators (Aristodemou & Tietze, 2018b).

[4]The area of the analysis visualisation tackles issues on the types of visualisation available,, how these can be improved and their effectiveness for different decisions. The expert focus group produced a mini-technology roadmap, which envisions an adaptive, interactive, intelligent, personalised search analysis with visualisation and interpretation.

[5]The area of patent quality is quite controversial and focuses on the many definitions and how it is measured (Squicciarini et al., 2013; Trappey et al., 2012). The experts of the focus group produced a mini-technology roadmap with a vision of transparency and inter-linkage of data, where there is the ability to match patents with products in a level playing field. An essential activity appears to be connecting different datasets and having multiple indicators. Ultimately, the integration of different data sources could lead to the availability of more data to determine patent quality. In terms of technology development, improvements in models using natural language processing, neural networks and deep learning approaches can better address the inclusion of both structured and unstructured data into the databases, with appropriate essential secure infrastructure.

[6]The experts agreed that the technology adoption of AI methods can be aided by more transparency and specialised data science training, with the aim that these analytic technologies stop being regarded as 'black box' solutions.

1.2 Thesis structure

The thesis is structured with a combination of the following: (i) the framework and methodology proposed by Ilevbare et al. (2016) for business methodology creation, and (ii) the decision support framework development methodology proposed by Turban et al. (2005).

Fig. 1.1 shows the overall approach followed in the structure of this thesis, the process we have followed, and the theoretical foundations of the research. Definition represents the research positioning (1.1.1), the future of patent analytics study (1.1.2), and the theoretical background (chapter 2)[1,2]. Computational resources are all the computational services we have used in developing this research. The data is stored in the cloud[3], and processed with virtual machines using Microsoft Azure[4] and Google AI Platform servers[5]. The code is written in Python language (Van Rossum & de Boer, 1991; Van Rossum & Drake, 1995, 2009), and is stored and maintained on GitHub[6].

Chapter 2 reviews the theoretical background of the research, with two main theoretical literatures: (i) the value of patents (section 2.1); (ii) the analysis of patent data using artificial intelligence, machine learning and deep learning methodologies, also defined as intellectual property intelligence (IPI) (section 2.2).

This is followed by chapter 3, which describes the development of the dataset: (i) the data collection (section 3.1); (ii) the data preparation (transformation) for AI deep learning methodologies (section 3.2), which includes the numeric, categoric and text feature representations. The dataset consists of all granted United States Patent and Trademark Office (USPTO) patents between 1976-2019.

Chapter 4 describes the development of the deep learning algorithmic approach using patent data for patent valuation. This includes: (i) the problem structure (section 4.1); (ii) the detailed analysis and representation of the network architecture (section 4.2); (iii) the evaluation metrics of the error-function derivative (section 4.3); (iv) the network optimisation

[1] All website URL links, referenced in the thesis, have been last effectively accessed on the 01.03.2021.

[2] All equations presented in the thesis follow the specific notation of the chapter they are enclosed. Chapter 2 contains equations about the feature definition, chapter 3 contains equations about the feature transformation, and chapter 4 contains equations about the development of the deep learning algorithm.

[3] Cloud computing is the on-demand availability of computer system resources or data centres, especially data storage and computing power, without direct active management.

[4] Microsoft Azure is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centers (Microsoft, 2020), URL: https://azure.microsoft.com/en-gb/

[5] Google AI Platform is a comprehensive machine learning service for developers and data scientists, which covers end-to-end spectrum of machine learning services including data preparation, training, tuning, deploying, collaborating and sharing of machine learning models (Google, 2020a), URL: https://cloud.google.com/ai-platform

[6] Github provides hosting for software development and version control using Git (Github, 2020).

with the parameter optimisation and error backpropagation algorithm (section 4.4); and (v) the deep learning implementation (section 4.5).

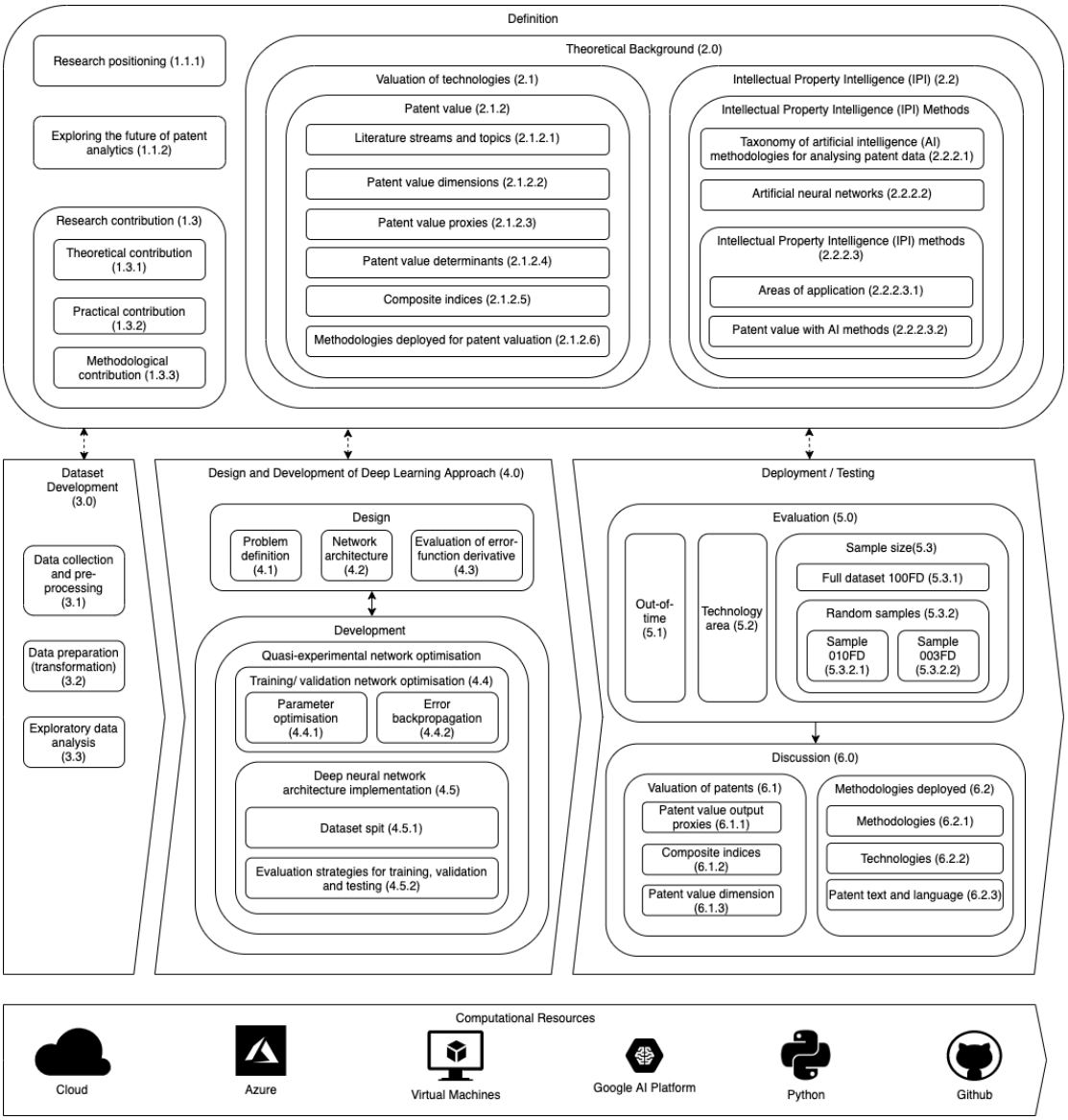


Fig. 1.1 Research approach for the development of the AI deep learning methodology for the value of patents

Chapter 5 presents the empirical results: (i) the out-of-time evaluation (section 5.1); (ii) the technology area evaluation (section 5.2); and (iii) the sample size evaluation (section 5.3). Chapter 6 discusses the results in two main directions: (i) in section 6.1 we discuss the results relative to the patent value literature, including proxies (6.1.1), composite indices (6.1.2), and value dimensions (6.1.3); (ii) in section 6.2, we compare the results to previous studies on intellectual property intelligence, for the methodologies deployed for the analysis

of patent data (6.2.1), technology forecasting (6.2.2), and the use of patent text and language (6.2.3). Chapter 7 concludes the research, with limitations and future research suggestions.

1.3 Contributions

This research provides the following contributions: (i) methodological (1.3.1), (ii) theoretical (1.3.2), and (iii) practical (1.3.3). All results and contributions should be interpreted with care, due to the sensitivity surrounding the value of patents and artificial intelligence (AI). Complementary contributions for further research and reproducibility also include: (i) the research code, which is available on GitHub[1], with the master branch representing the original code for the thesis; (ii) the research dataset, which is publicly available on the cloud[2].

1.3.1 Methodological contributions

This research makes some major methodological contributions (see 6.2). Primarily, this research proposes an AI deep learning methodology for the valuation of patents to identify valuable patents. Unlike previous studies on patent value that focus mostly on regression methodologies with samples with limited datapoints (Table 2.7), we propose an AI deep learning approach for patent valuation using big patent datasets. This is based on deep and wide artificial neural networks (ANN) of multi layer perceptrons (MLP), also known as deep learning, unlike some recent studies that concentrate on shallow ANNs with lagging output proxies (Table 2.12). We also make use of the cost-sensitive loss function, batch normalisation and L2 regularisation methods (see 4.4), allowing for improved learning and higher performance.

Furthermore, we also use and transform the patent text (abstract, claims, summary, title) into text vector features, using our developed Doc2Vec vector space model (VSM), which we use as input to our deep learning model. We evaluate all our models using a variety of strategies (out-of-time test, out-of-sample test, k-Fold and random split cross validation) (4.5), and transparently report all metrics (accuracy, confusion matrix, F1-score, false negative rate,

[1]The code is accessible from the repository of the author. The original time stamp of the master branch represents the code for the thesis, and is read-only. Any subsequent branches and merges, which have been modified by external researchers represents advancements of the methodology and build on material. Please, email the author to request access to the code (Aristodemou, 2020a).

[2]The dataset is accessible from the repository of the author. The original typestamp of the master dataset represents the dataset created and used for the thesis, and is read-only. Any subsequent dataset versions, which have been modified by external researchers represent alternative dataset versions and are saved separately. Please, email the author to request access to the dataset (Aristodemou, 2020b).

log loss, mean absolute error, precision, recall) (4.3). Thus, we advance the application of AI methods using patent data for patent valuation, including a text Doc2Vec VSM.

In addition, we advance the definition of *intellectual property analytics (IPA)*, defined by Aristodemou & Tietze (2018b), to *intellectual property intelligence (IPI)*. This is defined as the data science of analysing large amount of intellectual property (IP) information, specifically patent data, with artificial intelligence methodologies such as machine learning and deep learning, to discover relationships, trends and patterns in the data for decision making. For this definition, we move away from traditional analytical methods, and we focus on AI methods, which include the element of intelligence, i.e. the ability to acquire, learn and apply knowledge. This definition and the analysis of big patent data with AI-based approaches form the basis of the emerging field of IPI research and IPI studies.

Moreover, there are limited studies with large number of datapoints that mainly focus on categoric and numeric data, tailored to specific technology areas (Trappey et al., 2019, 2012). This research in particular, builds a large dataset, consisting of all granted USPTO patents from 1976-2019, with numeric, categoric and text features. The dataset is transformed into a vector space using methods such as one hot encoding (OHE) and Doc2Vec. This combination of all patent data features has improves the overall model performance. In addition, since this large dataset is not depended on the technology field, it has a wide applicability and higher generalisability, and is made publicly available for researchers to run efficiently and effectively run fairly complex data analysis (Aristodemou, 2020b).

1.3.2 Theoretical contributions

We explore a wide range of patent value output proxies, including a composite index, with deep learning and large datasets (6.1). Unlike current research (see 2.2.2.3.2), which focus on lagging proxies, we focus on a variety of output proxies. Specifically, we focus on 12 output proxies, which include grant_lag, generality, quality_index_4, and forward citations, generality_index and renewals in three time horizons (t4, t8, t12). We also explore the use of composite indices, such as the quality_index_4. In addition, we move away from relatively small datasets, limited to specific technological fields that reflect the characteristics of that area, to large generic datasets, improving the generalisability of the models.

Moreover, we associate the patent value proxies to the value dimension (see 6.1.3) they represent such as economic, strategic and technological (Frietsch et al., 2010). We observe that relying on a composite index, which takes into account a combination of dimensions yields higher results, because inherently the concept of economic growth has the elements of strategy and technology development (Teece, 1986). For economic value, we observe that early technology diversification (measured by the generality_index_t4), is important, yet in

later time horizons it becomes less important. This partly suggests that firms have wider technological competencies at the beginning to take advantage of new opportunities, which decrease with the increasing time horizon, focusing on core technologies (Hall, 2005; Hall & MacGarvie, 2010). In addition, for strategic value, we observe that as the time horizon increases, it becomes increasingly difficult to predict patent renewal, suggesting that firms maintain some of their earlier strategies and renew their patents in the fear of losing out (Granstrand, 1999).

1.3.3 Practical contributions

This research predict the value of patents with a variety of output proxies, including composite index proxies, for different technology areas (IPC classifications) and time horizons (see 6.2.2)[1]. Following the principles of technology roadmapping (Phaal, 2004), we use the models to value patents at different time horizons. This has also implications for technology management, and R&D management, with managers being able to use a variety of proxies to value patents at different stages of development. In the short term (t4), they could utilise together the models on citations_t4 and renewal_t4, for the medium term (t8), the models on citations_t8 and generality_t8, and for the long term (t12), the models on citations_t12 and renewals_t12, to forecast the value of their patented inventions, and subsequently of technologies.

From the technology area, technology managers could utilise the grant_lag model to forecast the value of patents in IPC E, the generality in IPC G, the quality_index_4, citations_t8, citations_t8, and renewal_t4 in IPC A, the citations_t4 and generality_t8 in IPC B, the generality_t4 in IPC C, the generality_t12 in IPC D, the renewal_t12 in IPC F, the generality in IPC G, and the renewal_t8 in IPC H. Thus, they can associate different output proxies per technology area when they are developing specific technologies, for example technologies related to physics or electrical, which can be useful for firms with limited resources, such as small-medium enterprises (SMEs). They can use Table 6.1 to factorise the grid of models to the particular case for the time horizon and technology area.

[1]The time horizon breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

Chapter 2

Theoretical Background

In this chapter, we focus on the theoretical background that underpins this research. It has been long argued that the value of technologies[1] (Mowery & Rosenberg, 1989; Rosenberg, 1994) can be modelled as the value of patented inventions[2].

We review the literature on the valuation of technologies and specifically, patent value (2.1.2). We focus on the patent value dimensions (2.1.2.2), proxies (2.1.2.3), determinants (2.1.2.4), and the methodologies deployed for patent valuation (2.1.2.6). We define patent value proxies as measures that can be used to approximate the value of patented inventions. They represent patent characteristics that have been used mainly in the literature as dependent variables and can be classified as ex-post[3]indicators (Lee et al., 2018; Noh & Lee, 2020; van Zeebroeck & van Pottelsberghe de la Potterie, 2011a). We define patent value determinants to represent patent characteristic that have been used mainly as value determinants, correlated with patent value. They represent patent characteristics that have been used in the literature as explanatory variables and can be classifed as ex-ante[3] indicators. Patent value proxies have also been used as patent value determinants in the literature (van Zeebroeck & van Pottelsberghe de la Potterie, 2011b).

[1]The definition of technology varies (Arrow & Intriligator, 2010). Verhoeven et al. (2016) define a technology as means to fulfil some purpose, which embodies principles and components that work in relation to each other to meet the purpose at hand. This is similar to the definition formulated by Bresnahan (2010) for general purpose technologies (GPT), and the technology form definition by Hall et al. (2010).

[2]There are many definitions of a patented invention, i.e. a patent. Nearly all agree that a patent is a temporary property right on an invention (Rockett, 2010), i.e. to exclude others from benefiting from the underlying intellectual property (Arora & Gambardella, 2010). It is an exclusive right granted for an invention, that provides a novel technical solution (The British Library, 2020; WIPO, 2020). Poege et al. (2019) argues that follows the principle of means to fulfil some purpose with the requirement that it contains at least one novel and inventive step.

[3]Patent characteristics can be classified into ex-ante and ex-post indicators (Arts et al., 2013). An ex-ante indicator is related to the nature of a patented invention, and is defined immediately at the point or just after the patent is filed. An ex-post indicator is related to the impact and value of a patented invention, which may change over time (Lee et al., 2018; Noh & Lee, 2020).

Following 1.1.2, we focus on the literature of the analysis of patent data with artificial intelligence (AI) methodologies, also defined as *intellectual property intelligence (IPI)* (Aristodemou & Tietze, 2018b). We review the IPI methods, by firstly forming a taxonomy of the methodologies (2.2.2.1), secondly, focusing on methodologies based on artificial neural networks (2.2.2.2), and finally, on the theoretical areas of application and the AI methodologies that have been applied with patent data (2.2.2.3).

2.1 Valuation of technologies

2.1.1 Background

The value of patented inventions varies widely at the patent, firm and industry levels (Thompson, 2010). The concept of patent value has acquired many meanings over time (Munari & Oriani, 2011; Parr & Sullivan, 1996; Pitkethly, 1997; So-Young et al., 2007). The many definitions that exist, i.e. technological value, economic value, are neither exclusive, nor do perfectly overlap, and users tend to bridge them into intuitive notions of value (Squicciarini et al., 2013). Recently, there has been much discussion about patent value, how to measure it and what it entails for innovation and technology development (Grimaldi & Cricelli, 2019).

The analysis of patent value plays an important role in managerial economic and business strategy because helps to estimate the value of the technologies. Various means are used to measure patent value according to different perspectives. Scholars have focused mainly on three research areas: (i) the economic value of patents based on information from patent databases and field surveys, (ii) firm value and performance, and (iii) the investigation of patent value determinants, sampling strategies and patterns. Under these broad research areas, scholars have investigated many topics, such as disruptive/ emerging technologies (Guderian, 2019), the geography of inventions (Adams, 2006), the globalisation of R&D activities (Narin & Hamilton, 1996), knowledge diffusion (Gambardella et al., 2007), patenting strategies (Granstrand, 1999), and technological performance (Aristodemou & Tietze, 2018a). Given the surge in world patenting (WIPO, 2019a,c, 2020), the added interest in the valuation of technologies, and the ability to analyse larger datasets of patents, we review the literature on the value of patents, to identify the patent value dimensions (2.1.2.2), proxies (2.1.2.3), determinants (2.1.2.4), and the methodologies deployed for patent valuation (2.1.2.6).

2.1.1.1 Review methodology

We aim to review the literature on patent value, to identify the methodologies deployed to value patents, the proxies and determinants used, and the patent value dimensions previous

literature focuses on. To carry out the review[1], the narrative and scoping literature review approaches have been adopted (Cronin et al., 2008; Paré et al., 2015), and a search strategy[2] has been developed (Robson, 2011). Fig.2.1 shows the review's process flow. The articles on patent value are identified from the Scopus database to find the most relevant published articles (Falagas et al., 2008). Focusing on recent literature, the search is constrained to articles published after 1989, to the fields of business, computer science, engineering, social science and mathematics. The core review identifies 93 articles. We then filter these by the Source-Normalised Impact Score (SNIP)[3] of the journal being greater than 1.00, to form a subset of 77 articles. For these, we review in detail the value dimensions, proxies, determinants and the methodologies deployed for patent value.

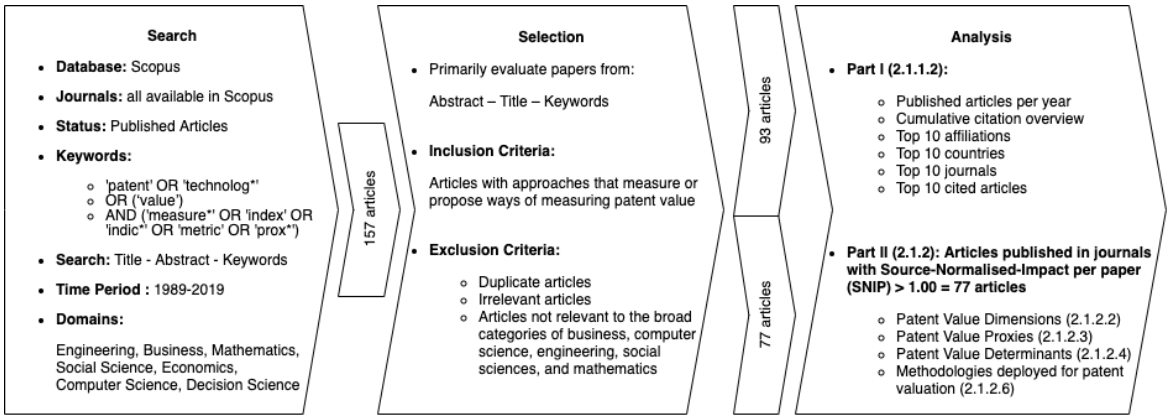


Fig. 2.1 Process flow of the review on the valuation of technologies

2.1.1.2 Bibliographic analysis results

The first level of the analysis focuses on the bibliographic information from the 93 articles (n_1). The number of articles have increased since 2017, reaching a peak of 13 articles

[1]The review draws predominantly on on-line resources (articles), and is supported by the narrative perspective using book references. This inherently introduces a gap in the literature review. We aim to reduce that gap using book and report references to complement our arguments and explanations.

[2]We search within the title, abstract and key words for various terms such as 'patent', 'technolog*'. The search is then narrowed to documents that also contain either in the title or the abstract or in the key words, the terms 'value', 'measure*', 'index', 'indicator*', 'indices', 'metric' and 'prox*'. We then form an initial set of articles from the 150 top cited articles, 100 most relevant articles by Scopus, and 150 articles from the top 10 publishing journals. The search is effective on 08.12.2019.

[3]Source Normalized Impact per Paper (SNIP) is a metric that intrinsically accounts for field-specific differences in citation practices, by comparing each journal's citations per publication with the citation potential of its field, defined as the set of publications citing that journal. SNIP therefore measures contextual citation impact and enables direct comparison of journals in different subject fields. A journal with a SNIP of 1.00 has the median number of citations for journals in that field. URL: https://www.elsevier.com/authors/journal-authors/measuring-a-journals-impact

published in 2019. Fig.2.2 shows the number of papers per year since 1989 (Fig. 2.2a), and the cumulative citation overview per article per year with more than 2 citations (Fig. 2.2b). There is an upward trend with the number of publications in recent years indicating an increasing interest in this particular field. There is a fairly even spread until 2017, and after there is an increase in the number of publications. In addition, the cumulative forward citations per year reach a peak in 2017 and then saturate.

Table 2.1 Top 10 affiliatons (2.1a), countries (2.1b), journals (2.1c), and cited articles (2.1d), for articles for valuation of technologies (n_1=93 articles)

(a) Affiliations (n_2=188 observations)

Affiliation[a]	No. of obs.	Share %
University of California, Berkeley, US	7	4%
KU Leuven, BE	6	3%
ULB Bruxelles, BE	5	3%
Cheongju University, KR	5	4%
CEPR, UK	5	3%
WHU - Otto Beisheim, DE	4	2%
Ludwig-Maximilians University, DE	4	2%
Universita di Cassino e del Lazio, IT	3	2%
Harvard University, US	3	2%
National Tsingua University, CN	3	2%
Total	40	21%

[a]Articles with one or more affiliations are multi-counted.

(b) Countries (n_3=71 observations)

Country[a]	No. of obs.	Share %
United States	26	21%
Germany	16	13%
United Kingdom	14	11%
Belgium	11	9%
Taiwan	9	7%
Italy	8	6%
China	7	6%
Australia	4	3%
Netherlands	4	3%
South Korea	4	3%
Total	103	82%

[a]Articles with one or more countries are multi-counted.

(c) Journals

Journal[a]	No.	Share %
Research Policy	22	24%
Scientometrics	13	14%
Technological Forecasting and Social Change	8	9%
World Patent Information	8	9%
Economics of Innovation	5	5%
International Journal of Innovation and Technology Management	3	3%
Journal of Intellectual Capital	3	3%
RAND Journal of Economics	2	2%
Review of Economics and Statistics	2	2%
Advanced Engineering Informatics	1	1%
Total	67	72%

[a]The 93 articles are published in 36 journals.

(d) Cited Articles

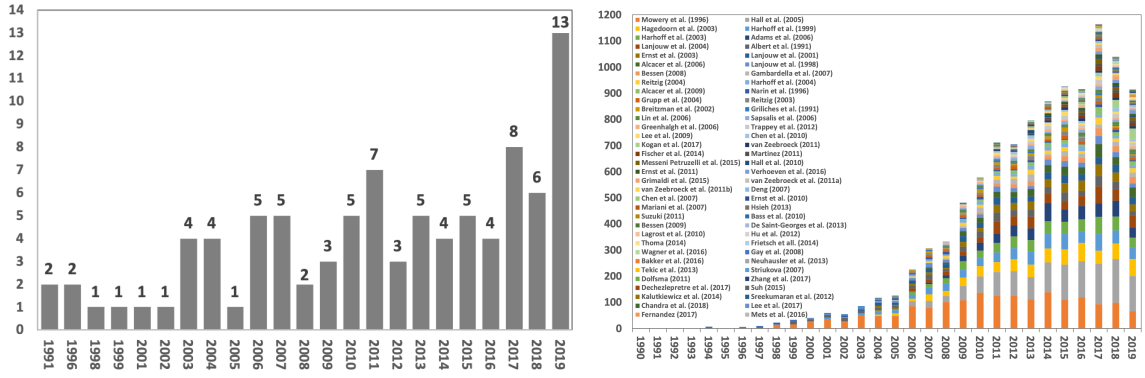
Article[a]	Citations	Cit. Freq.
Mowery et al. (1996)	1755	73.13
Hall (2005)	1304	86.93
Hagedoorn & Cloodt (2003)	623	36.65
Harhoff et al. (1999)	621	29.57
Harhoff et al. (2003)	538	31.65
Adams (2006)	516	36.86
Lanjouw & Schankerman (2004)	459	28.69
Albert et al. (1991)	429	14.79
Ernst (2003)	351	20.65
Lanjouw & Schankerman (2001)	349	18.37

[a]Citations frequency: total number of citations over the age of the article.

Table 2.1[4] shows the top 10 affiliations, countries, journals and citations of the 93 articles.

[4]In both Table 2.1a and Table 2.1b, any article with one or more affiliation from different countries is multi-counted (i.e. if an article has 3 different affiliations from 2 different countries, is counted 3 times in Table 2.1a and 2 times in Table 2.1b).

It is evident from the information that Europe is the leading continent. This is supported by Table 2.1b, where within the top 10 countries, 5 are in Europe (42% of the total share of observations). However, from Table 2.1b we observe that contributions are also made by the USA (21% share), since the majority of articles have been affiliated with the University of California, Berkeley (Table 2.1a). Moreover, the top 4 journals (accounting for 56%) are: Research Policy, Scientometrics, Technological Forecasting and Social Change, and World Patent Information. The top 10 journals account for 67 articles, indicating that articles in this field are concentrated within these journals (72% of the total share). In addition, the most cited articles are Mowery et al. (1996) with 1755 citations, followed by Hall (2005) with 1304 citations, and Hagedoorn & Cloodt (2003) with 623 citations. However, the article with the highest citation frequency[1] is Hall (2005) with 86.93, followed by Mowery et al. (1996) with 73.13.



(a) (b)

Fig. 2.2 Plots of: 2.2a Number of articles published per year (n_1=93) since 1989; 2.2b Cumulatitve citation overview per article per year, for articles with > 2 citations

2.1.2 Patent value

The analysis of patent value helps to estimate the value of the technologies (Grimaldi & Cricelli, 2019; Squicciarini et al., 2013). For example, Amazon's one-click patent[2] is considered one of the most valuable patents at the moment, since it enables a user to instantly place any purchasing ordering via any communications channel. This particular patent has been licensed widely to companies, including Apple, Facebook, and eBay. Patent value has several topics of investigation, in the research community (2.1.2.1). Given the surge in world patenting (WIPO, 2019a,c, 2020), the added interest in the valuation of technologies, and the

[1] Citation frequency is defined as the total number of citations over the age of the article.

[2] United States Patent with no. US 5,960,411, and title: Method and system for placing a purchase order via a communications network. The patent was granted in 1999 and is assigned to Amazon (Hartman et al., 1999).

ability to analyse larger datasets of patents, we review the patent value dimensions used by scholars (2.1.2.2), proxies (2.1.2.3), determinants (2.1.2.4), and the methodologies deployed to value patents (2.1.2.6).

2.1.2.1 Patent value literature streams and topics of investigation

Various means can be used to measure patent value[1,2]. It has been suggested that the patent value literature on patent value[3] can be organized into 3 broad research streams (van Zeebroeck & van Pottelsberghe de la Potterie, 2011b). Table 2.2 provides an overview of the research streams and the relevant theoretical topics.

The first research stream focuses on the estimation of the economic value of patents based on information from field surveys or patent databases or transactions (Tietze, 2012). The most well known field survey is the PatVal study, where European inventors were surveyed to estimate retrospectively the value of their inventions, after the patent expiration (Gambardella et al., 2007, 2005)[4]. These studies find that the patent value distribution is positively skewed, with a long right tail, and the majority of patents having little value (Deng, 2007; Gambardella et al., 2005; Hall, 2005).

The second research stream focuses on using intellectual property rights (most notably patents) to analyse the impact of innovation on firm value and performance. Several scholars correlate patent value determinants with firm value (Lanjouw & Schankerman, 2004). Hall

[1]The value of a patent is rarely observable and inductive approaches are usually used (Harhoff et al., 2003; Reitzig, 2003). One could find a direct correlation with observable prices or operationalise latent determinants of patent value, which correlate with observable effects or proxies (Reitzig, 2004). The nature of value recognition has two dimensions: the intrinsic and extrinsic. The intrinsic dimension of the patent value is derived from the intrinsic technological significance, and is represented by all that appear in the patent document, where as the extrinsic value is the potential to develop the market (Grimaldi & Cricelli, 2019).

[2]With the focus on the analysis of patent data (see 1.1.2), we can also define *intelligent patent value (IPV)* as the value of a patent arising or being represented by patent data determinants and proxies, i.e. arising from its technological significance (value of the invention) with the potential to develop a market (patent premium) on dimensions such as economic, technological, social and strategic (see 2.1.2.2), and which is measured or analysed by artificial intelligence (AI) and data science methodologies (Aristodemou & Tietze, 2018b).

[3]The value of patent also consists of two parts: (i) the patent value related to the market protection given by the patent, i.e. the value of the patent rights, and (ii) the value of the invention, i.e. the value to the firm without information disclosed within the patent being released (Arora et al., 2008; Arora & Gambardella, 2010; Jensen et al., 2011; Pitkethly, 1997; Thoma, 2016). The patent premium represents the value of the patent related to the market protection. Arora et al. (2008) argue that patenting firm's expected earnings are between 75-125% more than by not disclosing it. Patent premium's determinants include, but are not limited to, the industrial sector, competition dynamics, regulation impact etc. (Greenhalgh & Rogers, 2006; Harhoff et al., 2003).

[4]We are aware that the PatVal study has estimates of patent value (Gambardella et al., 2005). Unfortunately, it was not possible to source the PatVal dataset despite numerous trials and communication channels. Given the difficulty in sourcing this dataset, we focus on alternative proxies capable of estimating patent value, of which some correlate with the patent premium because of patent disclosure (Arora & Gambardella, 2010; Jensen et al., 2011).

et al. (2005) explore the use of patents citations as a measure of patent importance, and associate them to firm's stock market value.

Table 2.2 Patent value literature research streams and theoretical topics

(a) Literature streams

	Literature Stream[a]	Authors (examples)
1	Economic value of patents based on information from patent databases and field surveys	Bessen (2008); Ernst (2003); Gambardella et al. (2007); Harhoff et al. (1999); Harhoff & Reitzig (2004); Harhoff et al. (2003); Lanjouw & Schankerman (2001); Lanjouw et al. (1998); Reitzig (2003, 2004)
2	Firm value and performance	Bessen (2009); Greenhalgh & Rogers (2006); Hall (2005); Hall & MacGarvie (2010); Lanjouw & Schankerman (2004)
3	Investigation into determinants, sampling strategies and patterns in relation to patent value	Adams (2006); Albert et al. (1991); Alcácer et al. (2009); Aristodemou & Tietze (2018b); Bass & Kurgan (2010); Ernst & Omland (2011); Frietsch et al. (2010); Griliches et al. (1991); Grimaldi et al. (2015, 2018); Grupp & Mogee (2004); Mowery et al. (1996); Narin & Hamilton (1996); van Zeebroeck (2011); van Zeebroeck & van Pottelsberghe de la Potterie (2011b)

[a]The list of literature streams is not exhaustive. We aim to give an overview of the broad literature streams and the purpose of researching patent value.

(b) Theoretical topics of investigation

Topic[a]	Definition	Authors (examples)[b]	Total articles	<2010[c]	2011-2018	2018-2020
Disruptive/ Emerging Technologies	Patents are used to detect progression of technologies.	Berg et al. (2018); Guderian (2019); Mariani et al. (2019); Trappey et al. (2012)	4	0	1	3
Economic value of inventions	The economic value of inventions is important for its economic impact	Albert et al. (1991); Hall (2005); Harhoff et al. (1999, 2003); Lanjouw et al. (1998); Reitzig (2003, 2004)	30	16	11	3
Geography of Invention	Patents hold information on the region, country, inventor and applicant details	Adams (2006); Bessen (2008); Gambardella et al. (2007); Hagedoorn & Cloodt (2003); Narin & Hamilton (1996)	6	5	0	1
Globalisation of R&D activities, and role of universities	Patents include information on the inventive performance and activities of multinational firms; it's possible to track the patterns and the intensity of international cooperation	Bessen (2008); Ernst (2003); Ernst & Omland (2011); Frietsch et al. (2014); Greenhalgh & Rogers (2006); Mariani & Romanelli (2007); Narin & Hamilton (1996); Sapsalis et al. (2006)	15	8	6	1
Knoweldge diffusion	Patents provide detail description on prior art, and can identify the influence and progression of technologies	Alcácer & Gittelman (2006); Chandra & Dong (2018); Gambardella et al. (2007); Hagedoorn & Cloodt (2003); Mowery et al. (1996); Wang (2015)	7	4	2	1
Patenting strategies	Patents reveal the timeline of inventions, which identifies the market strategy of the patent owner	Breitzman & Mogee (2002); Dolfsma (2011); Ernst (2003); Ernst & Omland (2011); Gambardella et al. (2007)	8	3	5	0
Performance and mobility of researchers	Patents have a variety of information including countries and sectors, which can be used to measure innovation performance	Alcácer et al. (2009); Greenhalgh & Rogers (2006); Grupp & Mogee (2004); Lanjouw & Schankerman (2004); Narin & Hamilton (1996)	5	5	0	0
Technological performance	Patents are used to monitor the technological performance and track the technological positioning of companies, regions or countries	Aristodemou & Tietze (2018b); Grimaldi et al. (2015); Grupp & Mogee (2004); Hagedoorn & Cloodt (2003); Harhoff & Reitzig (2004); Lanjouw & Schankerman (2001); Veugelers & Wang (2019)	17	3	9	5

[a]The topics and definitions have been compiled using the taxonomy suggested by OECD (2009). Many articles belong to more than one topic under investigation.

[b]We provide a few examples of studies for every topic. The list is not exhaustive and the aim is provide an overview of the research topics.

[c]The year 2010 is indicative of when popularity around artificial intelligence (AI) methodologies has increased due to advancements in compuer science (Murphy, 2012; Schmidhuber, 2015).

Drawing on the above, the third research stream investigates patent value latent determinants, internal patterns and relationships, in relation to patent value (Mowery et al., 1996; van Zeebroeck, 2011). Several scholars identify different patent value latent determinants, as explanatory variables, which are correlated to independent value measures (van Zeebroeck & van Pottelsberghe de la Potterie, 2011b).

We then focus on the theoretical topics of investigation. Table 2.2b[1] shows the theoretical topics of investigation associated with patent value, with the topic, definition and some examples of studies The majority of articles are found under the topic of economic value of inventions, followed by the topics on technological performance and globalisation of R&D. These have been published before 2010. Topics that have gained attention since 2011 include the technological performance, patenting strategies, and disruptive/ emerging technologies. The topic of the economic value of inventions has maintained interested, throughout time.

2.1.2.2 Patent value dimensions

We aim to identify the patent value dimensions previous studies focus on to give an overview of the discussions around patent value. Table 2.3 gives an overview of the 4 patent value dimensions, economic, strategic, social and technological, identified in the literature, together with their definitions and examples of previous studies. It is possible to evaluate patent value on different value dimensions. Table 2.3 reveals that the majority of the articles are around the economic value dimension. The economic value of a patent is defined as the degree to which the patent enters or creates a new market, or the patent's private asset value (Grimaldi et al., 2018). This is a generally agreed definition within the innovation management and economics literature (Deng, 2007; OECD, 2009).

The technological value dimension is defined as the degree to which a patent contributes to further developing advanced technology (Frietsch et al., 2010), and the implementation of successive technologies (Chandra & Dong, 2018). Despite a few articles researching the topic of patent strategies (Table 2.2b), the strategic value dimension has been investigated by a limited number of articles. It is defined as the degree to which the patent is used strategically, with underlying strategic motives (Frietsch et al., 2010). Ernst (2003) provides a conceptual framework on how strategic value can be realised. The least identified dimension is the social value, which is defined as the degree to which a patent impact society by collectivity and altruism (Frietsch et al., 2010; Gay et al., 2008)[2].

[1] We construct the table by reviewing the aim and purpose of the studies identified (in 2.1.1.1), and then we categorise them in the topics. These topics and their definitions are identified in the taxonomy published by OECD (2009).

[2] There has been a long discussion in the literature about the economic value of patents, with several methods and approaches being proposed (Munari & Oriani, 2011; Parr & Sullivan, 1996; Pitkethly, 1997;

Table 2.3 Patent value dimensions

Patent Value Dimension[a,b]	Definition	Authors (examples)[c]
Economic Value (EconV), including market-based constructs such as firm value and estimated monetary patent value)	Degree to which the patent enters or creates a new market, or its private asset vale once realised in terms of sale price, royalties, licenses, costs (including but not limited to R&D costs, litigations costs, etc.) (Frietsch et al., 2010; Lagrost et al., 2010)	Bessen (2008, 2009); Deng (2007); Ernst (2003); Frietsch et al. (2014); Gambardella et al. (2007); Gay et al. (2008); Hall (2005); Hall & MacGarvie (2010); Harhoff et al. (1999); Harhoff & Reitzig (2004); Kapoor et al. (2013); Lanjouw et al. (1998); Lanjouw & Schankerman (2004); Mariani & Romanelli (2007); Mowery et al. (1996); Striukova (2007)
Strategic Value (StrV)	Degree to which the patent is used strategically, with underlying strategic motives to, but not limited to, blocking competitors, easier access to financial markets, preventing key technologies from being invented around and the generation of licensing revenues (Frietsch et al., 2010; Granstrand, 1999)	Ernst (2003); Hsieh (2013)
Social Value (SocV)	Degree of wich a patent impacts society by collectivity and altruism (Frietsch et al., 2010)	Frietsch et al. (2010); Gay et al. (2008)
Technological Value (TechV)	Degree to which a patent contributes to further developing advanced technology (Frietsch et al., 2010). Technical value is a subset of technological value. It is defined as the degree to which the practical realisation of the technology described by the patent at a commercial scale is realised, and is revealed through the importance of the patent to the implementation of successive technologies (Chandra & Dong, 2018).	Albert et al. (1991); Aristodemou & Tietze (2018b); Bekkers et al. (2011); Chandra & Dong (2018); Khachatryan & Muehlmann (2019); Kogan et al. (2017); Lanjouw & Schankerman (2004); Lee et al. (2017); Mowery et al. (1996); Suzuki (2011); Verhoeven et al. (2016)

[a]The list is not exhaustive and provides a summary of the identified dimensions in the literature. It includes the dimension's definition and some examples.

[b]The value dimensions are arranged in alphabetical order.

[c]Some articles might belong to more than one dimension.

2.1.2.3 Patent value proxies

From the review, we focus on patent value proxies[3] to identify the most frequently used, and their association to patent value dimensions (Table 2.3), with the aim to understand what previous scholars have used to approximate the value of a patented invention. These proxies can represent a variety of patent value dimensions, and should be interpreted with care. Table 2.4 provides an overview of the most frequently used proxies in the literature in alphabetical order, with their definition, rational, and some examples of studies. We associate them to their respective patent value dimension (Table 2.3) and provide some criticism on how they have been used in the text (Grimaldi et al., 2018; OECD, 2009).

So-Young et al., 2007). We focus on the definition by Deng (2007), also supported by the OECD (2009), which narrowly defines the economic value as the degree which the patent enters or creates a new market (Arora & Gambardella, 2010). Given that we focus on patent data, we take an inductive approach towards the intrinsic economic value of patents and how this can be reflected from patent data only by operationalising latent determinants and proxies (Grimaldi & Cricelli, 2019; Reitzig, 2004; Squicciarini et al., 2013). The value of patent also consists of two parts: (i) the patent value related to the market protection given by the patent, i.e. the value of the patent rights, and (ii) the value of the invention, i.e. the value to the firm without information disclosed within the patent being released (Arora et al., 2008; Arora & Gambardella, 2010; Jensen et al., 2011; Pitkethly, 1997; Thoma, 2016). The patent premium represents the value of the patent related to the market protection.

[3]We use the term patent value proxy to represent patent characteristics that have been been used mainly in the literature as dependent variables (van Zeebroeck & van Pottelsberghe de la Potterie, 2011a). We use the term patent value determinants to represent patent characteristic that have been used mainly as explanatory variables (van Zeebroeck & van Pottelsberghe de la Potterie, 2011b). Patent value proxies have also been used as patent value determinants.

2.1.2.3.1 Forward citations

Forward citations have been used widely to assess the technological impact and economic value of inventions (Aristodemou & Tietze, 2018b). Patents receiving citations indicate the existence of downstream research efforts. The value of a patent and the number of its forward citations are positively correlated (Hall, 2005; Harhoff et al., 1999; Reitzig, 2004; Trajtenberg, 1990). Scholars have also argued that patents receiving more citations are likely to be renewed (Bessen, 2008; Lanjouw et al., 1998). Hagedoorn & Cloodt (2003) study the innovative performance of large firms using a variety of proxies including forward citations. Lanjouw & Schankerman (2001) look into the cost of engaging in litigation, and they argue that the probability of litigation diminishes the patent value as a research investment incentive.

A major criticism of this proxy is that it is a lagging indicator, i.e. only becomes realised with time accumulation (Choi et al., 2020). The number of citations is truncated because only citations at any point in time are known (Squicciarini et al., 2013). It is influenced by differences in patent examination practises across time and patent offices, and the patent's technological area (OECD, 2009). Thus, scholars relying only on forward citations to measure the value of patents should be careful when assessing their results.

2.1.2.3.2 Generality index

The generality index calculates the diversification of the technological classes distribution in the forward citations (Trajtenberg et al., 1997), i.e. the range of technology fields that cite the patent, and it's high if the forward citations of a patent belong to a wide range of fields (Squicciarini et al., 2013). Several scholars have used the proxy to identify general purpose technologies or to identify high market value technologies (Chen & Chang, 2010d). Wagner & Wakeman (2016) investigate the relationship between patent indicators that capture patent value to the outcomes of the product development process. They use the generality index to capture a patent's diversified impact to a variety of technological fields.

One of the major criticism of the proxy, is that it treats technologies that are closely related but are not in the same class in the same way as they treat very distant technology fields. Due to its variable definition, it's not easily comparable since scholars have defined the *n*-digit IPC technology classes differently.

2.1.2.3.3 Grant lag

The time it takes for a patent to be examined, between the filing date of the application and the date of the grant, is defined as the grant lag, which is correlated to patent value (Squicciarini et al., 2013). A granted patent signals legal protection of the underlying patented invention, and a proxy of value (OECD, 2009).

Table 2.4 Patent value proxies

Proxy[a,b]	Dimension[c]	Definition[d]	Rational[e]	Authors (examples)[f]
Forward Citations (2.1.2.3.1)	TechV, EconV	The number of citations a given patent receives (forward citations) represents the technological importance of the patent for the development of subsequent technologies (Aristodemou & Tietze, 2018b). $$FC_{i,T} = \sum_{t=P_i}^{P_i+T} \sum_{j \in J(t)} Cit_{j,i} \quad (2.1)$$ where $FC_{i,T}$ is the number of forward citations received by patent i granted in year P_i within T years from its grant date. $Cit_{j,i}$ is variable that equals 1 if the patent document j cites patent document i, and 0 otherwise. $J(t)$ is the set of all patents applications published in year t.	The number of citations a patent receives shows the economical and technological importance of a patent (Aristodemou & Tietze, 2018b; Hall, 2005).	Albert et al. (1991); Alcácer & Gittelman (2006); Bessen (2008); Breitzman & Mogee (2002); Frietsch et al. (2010); Hagedoorn & Cloodt (2003); Hall & MacGarvie (2010); Harhoff et al. (1999); Harhoff & Reitzig (2004); Harhoff et al. (2003); Lanjouw & Schankerman (2001, 2004); Lin et al. (2006); Mowery et al. (1996); Reitzig (2004); Sapsalis et al. (2006); Trajtenberg (1990)
Generality Index (2.1.2.3.2)	TechV, EconV	The patent generality index is based on a modified Hirschman-Herfindahl Index (HHI) (Aristodemou & Tietze, 2018b). It calculates the diversity of the distribution of forward citations in technology classes (n-digit IPC technology classes), with a range 0-1. It's high if the forward citations of a patent belong to a wide range of fields (i.e. the patent is relevant for later inventions, and not only in its own technology class) (Squicciarini et al., 2013). $$G_X = 1 - \sum_{j=1}^{M_i} \left(\frac{1}{N} \sum_{i=1}^{N} \frac{T_{ji}^n}{T_i^n} \right)^2 \quad (2.2)$$ where X is the focal patent, Y_i patents citing the focal patent X, T_j^n is the total number of IPC n-digit classes in y_i, T_{ji}^n is the total number of IPC n-digit classes in the j^{th} IPC-k-digit class in y_i, $k = 1...8$ is the hierarchy level of the IPC class, $j = 1...M_i$ is the cardinal of all IPC-k-digit classes in y_i.	Hall et al. (2001) discover the existense of the relationship between the generality index and the value of patents, where a higher generality index (higher diversification of technology classes in forward citations) demostrates a higher value of the patented invention	Chen & Chang (2010d); Duguet & MacGarvie (2005); Gay et al. (2008); Grimaldi et al. (2018); Hall et al. (2001); Hall (2005); Harhoff et al. (2003)
Grant Lag (2.1.2.3.3)	StrV	A granted patent signals a strategic legal protection of an underlying patented invention. The time elapsed between the filing date of the application and the date of the grant, is defined as the grant lag index, which is correlated to the value of the patent. $$GL_{Pi} = \Delta t \quad (2.3)$$ where for each patent P, is the number of days elapsing between patent application date and patent granting date.	Harhoff & Wagner (2009) argue that applicants try to accelerate the grant procedure for their most valuable patents (i.e. by well documenting their application)	Grimaldi & Cricelli (2019); Harhoff et al. (2007); Harhoff & Reitzig (2004); Ma et al. (2019); Squicciarini et al. (2013); Thoma (2014); van Zeebroeck & van Pottelsberghe de la Potterie (2011a)
Renewals (2.1.2.3.4)	EconV, StrV	The renewal of a patent signals that the patented invention has value for the owner. $$Re_{i,T} = \begin{cases} 1 \\ 0 \end{cases} \quad (2.4)$$ where $Re_{i,T}$ is the renewal ($= 1$) of a granted patent i, after T years from its grant date, where $T = \{4, 8, 12\}$.	Patent renewals represent the owner's rational to make profit maximising renewal decisions (OECD, 2009)	Bakker (2017); Bessen (2008); Deng (2007); Harhoff et al. (2003); Lanjouw et al. (1998); Liu et al. (2008); Tahmooresnejad & Beaudry (2018); Thoma (2014); Thompson (2017); van Zeebroeck (2011); van Zeebroeck & van Pottelsberghe de la Potterie (2011a); Wang (2015)

[a] Patent value proxies are measures that can be used to approximate the value of patented inventions. They represent patent characteristics that have been used mainly in the literature as dependent variables and can be classified as ex-post indicators (van Zeebroeck & van Pottelsberghe de la Potterie, 2011a).

[b] Proxies are arranged in alphabetical order, and refer to 2.1.2.3.

[c] Patent value dimension is defined according to Table 2.3.

[d] Definition includes a description of the proxy and its formula.

[e] Rational refers to the purpose of the proxy and how it is interpreted.

[f] Studies might belong to more than one proxy because they make use of more than one. The list is not exhaustive, and a selection is shown here.

Harhoff & Wagner (2009) discover the relationship between the value of a patent and length of the grant lag period, where they argue that applicants try to accelrate the grant procedure (by having a shorter grant lag index) for their most valuable patents. However, the grant lag is subject to the examination process, which varies depending on the patent office, the examiner, and the filing time (van Zeebroeck & van Pottelsberghe de la Potterie, 2011a). Thus, it is subject to changes in the examination process. In addition, the process also depends on the applicant and how fast they action the examination report results.

2.1.2.3.4 Renewals

The renewal of a patent signals that the patent is still useful and has some value to the owner. A patent holder pays a maintenance fee immediately after a patent is granted to retain exclusive rights, and needs to keep paying this fee in years 4, 8 and 12[1]. Information about the renewal of patents has been used in a number of studies, which have generally suggested that more valuable patents are renewed for longer periods (Gambardella et al., 2007). Scholars have used patent renewal data to estimate the private value of patents, based on the rational that owners make profit maximising renewal decisions (Bessen, 2008). Lanjouw et al. (1998) also find that renewals are correlated with the value of patents and larger patent families. This proxy also suffers from criticism, since the renewal of patents is mainly administrative and is depended on patent offices. Sometimes due to the lack of information and communication between the patent office and the patent holder, the fees are not paid in time, and this delays the patent status.

2.1.2.4 Patent value determinants

Following the review of the proxies in 2.1.2.3, we focus on patent value determinants[2] to identify the most frequently used determinants, and their explanatory association to patent value dimensions. We aim to understand what previous scholars have used. The determinants can represent a variety of patent value dimensions, and should be interpreted with care. Table 2.5 provides an overview of the most frequently used determinants in alphabetical order, with their definition, rational, and some examples of studies that have used it. We associate them to their respective patent value dimension (Table 2.3 and then provide some criticism in the text on how they have been used (Grimaldi et al., 2018; OECD, 2009).

[1] USPTO (2020), URL: https://www.uspto.gov/patents-maintaining-patent/maintain-your-patent

[2] We use the term patent value determinants to represent patent characteristic that have been used mainly as explanatory variables (van Zeebroeck & van Pottelsberghe de la Potterie, 2011b).

2.1.2.4.1 Backward citations

Backward citations are used to assess an invention's patentability defining the legitimacy of the claims (OECD, 2009), and help to track knowledge spillovers in technology. Harhoff & Reitzig (2004) show how patent characteristics influence the probability of opposition. Their analysis shows that backward citations are positively correlated to the probability of oppositions. Hall (2005) find a strong correlation between patent value and backward citations, similar to Sapsalis et al. (2006) who find that the identification of the prior art allows for an improved understanding of the value determinants (Chandra & Dong, 2018). Backward citations are a static determinant, defined at one point in time and very rarely adjusted. It depends on the examination process and the examiner, and many times firms try to cite a lower number of backward citations.

2.1.2.4.2 Claims

Claims represent the scope of a patent, which is an important determinant of its economic value. It defines the legal dimensions of protection and the extent of market power attributed to the patent. A broader scope refers to a broader technology area from which others are excluded. The number and content of the claims determine the scope and breadth of the patent rights (OECD, 2009). This also applies for all patent text sections (abstract, claims, summary, title) with only limited scholars operationalising it to textual feature representation (Wang & Chen, 2019). Lanjouw & Schankerman (2004) find that the number of claims is the most important determinat of the quality of patents. Shane (2001) argue that only highly valued patents, underpinned by several technical claims, increase the firm's market value.

One criticism is that the structure of the patent fee is generally based on the number of claims contained in the document, a large number of claims might also imply higher fees (Lanjouw & Schankerman, 2001). In addition, Lanjouw & Schankerman (2001) also find that the probability of litigation increases with the number of claims. There are changes in the number of claims, and its contents, according to the patent technology field (Ernst, 2003; Squicciarini et al., 2013), and the assignee firm.

2.1.2.4.3 Family size

A patent family can be defined as a set of patents filed in several countries which share at least one common priority filing, through the Patent Cooperation Treaty (PCT) (Martínez, 2011). The size of patent family, represented by the number of countries or number of jurisdictions, has been found to be correlated to patent value (Lanjouw & Schankerman, 2001; Lanjouw et al., 1998). Large international patent families have been found to be particularly valuable (Deng, 2007; Gambardella et al., 2007; Harhoff et al., 2003; Reitzig, 2004; van Zeebroeck

& van Pottelsberghe de la Potterie, 2011a). Some scholars have used the patent family size to represent the extent of patent protection in global markets, i.e. market coverage (Ernst & Omland, 2011; Lagrost et al., 2010; Lanjouw & Schankerman, 2004; Squicciarini et al., 2013; van Zeebroeck & van Pottelsberghe de la Potterie, 2011a). One of the major criticism is that there are many patent family definitions, which makes the patent counting quite complicated (Adams, 2006; Martinez, 2010). Every patent office and patent data provider use different patent family definitions, which make the identification of priority filings difficult.

2.1.2.4.4 Non-Patent Literature (NPL) references

Non-patent literature (NPL) citations can be considered as indicators of the contribution of public science to industrial technology (Narin & Hamilton, 1996). Patents that cite science may contain more complex and fundamental knowledge, and the number of NPL citations are correlated to the value of patented invention (Messeni Petruzzelli et al., 2015; Reitzig, 2004). They provide insights into technologies closer to scientific knowledge (Gay et al., 2008).

2.1.2.4.5 Originality index

The originality index operationalises the concept of knowledge diversification, i.e. breadth. Inventions relying on a large number of diverse knowledge sources lead to patents belonging to a wide array of technology fields (Harhoff & Wagner, 2009; Trajtenberg et al., 1997). Bessen (2008) use it as a value determinant to estimate the value of US patents based on patent renewal data (Dolfsma, 2011; Lin et al., 2006). It is a static indicator, and rarely changes at the sector and country level (Squicciarini et al., 2013).

2.1.2.4.6 Radicalness index

The radicalness index measures the technological radicalness, by counting the number of unique IPC technology classes of the backward citations, not including the classes of the focal patent (Shane, 2001). The higher it is (max=1), the more the invention should be considered radical, building upon technology fields other than the one it is applied (Grimaldi & Cricelli, 2019; Mariani et al., 2019; Squicciarini et al., 2013).

2.1.2.4.7 Scope

The scope of patents is defined as the number of technical classes (as indicated by the number of IPC/ CPC classes) attributed to a patent, and is associated to patent value (Lanjouw et al., 1998). A criticism of this determinant is that the scope can change, following a court decision, or adjusted at the filing stage according to an assignee and the patent office.

Table 2.5 Patent value determinants

Determinant[a,b,c]	Dimension[d]	Definition[e]	Rational[f]	Authors (examples)[g]
Backward Citations (2.1.2.4.1)	TechV, EconV	They are prior knowledge on which a focal patent application has relied on, and are used to assess an invention's patentability (OECD, 2009) $$BC_{i,T} = \sum_{t=P_i} \sum_{j \in j(t)} Cit_{j,i} \quad (2.5)$$ where $BC_{i,T}$ is the no. of backward citations of focal patent i granted in year P_i. $Cit_{j,i}$ equals 1 if the patent j is cited by focal patent i, and 0 otherwise. $J(t)$ is the set of all patents applications published in year t.	The no. of backward citations assess the degree of novelty of an invention (Criscuolo & Verspagen, 2008; Lanjouw & Schankerman, 2001)	Chandra & Dong (2018); Fischer & Leidinger (2014); Frietsch et al. (2014); Grimaldi et al. (2018); Harhoff & Reitzig (2004); Lanjouw & Schankerman (2004); Lee et al. (2017); Mariani & Romanelli (2007); Messeni Petruzzelli et al. (2015); Reitzig (2004); Sapsalis et al. (2006); Thoma (2014); Trappey et al. (2012)
Claims (2.1.2.4.2)	TechV, EconV	They determine the boundaries of the exclusive rights of a patent owner, given that only the technology covered in the claims can be legally protected and enforced. *Variety:* no. of claims, no. of independent/ dependent/ process/ application claims, Text (Abstract, Claims, Tittle, Summary).	The number and content of the claims determine the scope and breadth of the patent rights (OECD, 2009).	De Clercq et al. (2019); Grimaldi et al. (2015); Lanjouw & Schankerman (2004); Marco et al. (2019); Milanez et al. (2017); Porter et al. (2019); Thoma (2014); Trappey et al. (2012); Zhang et al. (2017)
Family (2.1.2.4.3)	EconV, StrV	A patent family can generaly be defined as a set of patents filed in several countries which share at least one common priority filing (Lanjouw & Schankerman, 2001, 2004; Martínez, 2011). *Variety:* no. of family members, Countries of family members, patent family definition	The size of family (no. of jurisdictions) is correlated to patent value (Lanjouw et al., 1998).	Chandra & Dong (2018); Deng (2007); Fischer & Leidinger (2014); Gambardella et al. (2007); Gay et al. (2008); Grimaldi et al. (2018); Harhoff et al. (2003); Reitzig (2004); Sapsalis et al. (2006); Trappey et al. (2012)
Non-Patent Literature (NPL) (2.1.2.4.4)	TechV	NPL citations are a list of prior art references to scientific papers that set the boundaries of patents' claims for novelty, inventive activity and industrial applicability, and consists of peer-reviewed scientific papers, conference proceedings, databases and other relevant literature. *Variety:* no. NPL citations, vector embeddings of NPLs	NPLs can be considered as indicators of the contribution of science to industrial technology.	Bass & Kurgan (2010); Chandra & Dong (2018); Fischer & Leidinger (2014); Gay et al. (2008); Harhoff & Reitzig (2004); Harhoff et al. (2003); Narin & Hamilton (1996); Reitzig (2004); Suzuki (2011); Veugelers & Wang (2019)
Originality Index (2.1.2.4.5)	TechV	It refers to the breadth of the technology fields on which a patent relies, i.e. inventions relying on diverse knowledge sources lead to patents belonging to a wide array of technology fields. $$Orig_X = 1 - \sum_{j=1}^{M_i} \left(\frac{1}{N} \sum_{i=1}^{N} \frac{T_{ji}^n}{T_i^n} \right)^2 \quad (2.6)$$ where X is the focal patent, Y_i patents cited (backward citations) by the focal patent X, T_j^n is the total number of IPC n-digit classes in y_i, T_{ji}^n is the total number of IPC n-digit classes in the j^{th} IPC-k-digit class in y_i, $k = 1...8$ is the hierarchy level of the IPC class, $j = 1...M_i$ is the cardinal of all IPC-k-digit classes in y_i.	It measures knowledge diversification (Trajtenberg et al., 1997)	Bessen (2008); Gay et al. (2008); Grimaldi & Cricelli (2019); Lin et al. (2006); Suh (2015)
Radicalness Index (2.1.2.4.6)	TechV, EconV	It measures the technological radicalness of inventions, by counting the number of IPC technology classes in which the patents cited by the focal patent are, but in which the focal patent is not classified (Shane, 2001). $$Rd_P = \sum_{j}^{n_p} \frac{CT_j}{n_p}; IPC_{pj} \neq IPC_p \quad (2.7)$$ where CT_j denotes the count of IPC-k-digit codes IPC_{pj} of patent j cited in patent p that is not allocated to patent p, out of n IPC classes in the backward citations counted at the most disaggregated level available.	The higher the ratio, the more diversified the array of technologies on which the patent relies upon, the more the invention should be considered radical	Grimaldi & Cricelli (2019); Shane (2001); Squicciarini et al. (2013)
Scope/ Technological (IPC, CPC) Classification (2.1.2.4.7)	TechV, EconV	The scope is associated with patent value, and represents the protection regime. $$SCp = n_p \quad (2.8)$$ where n_p denotes the no. of distinct k digit IPC classifications. *Variety:* categorical representation of IPC/ CPC classifications, no. of primary/ addition IPC/ CPC classifications	Higher no. classifications indicate higher scope, and thus patent value	Alcácer & Gittelman (2006); Gambardella et al. (2007); Harhoff & Reitzig (2004); Lanjouw & Schankerman (2001); Reitzig (2004); Trappey et al. (2012); Verhoeven et al. (2016); Zhang et al. (2017)

[a] Patent value determinants represent patent characteristic that have been used as explanatory variables and can be classifed as ex-ante indicators (Lee et al., 2018; Noh & Lee, 2020; van Zeebroeck & van Pottelsberghe de la Potterie, 2011a).

[b] This is not an exclusive list of patent value determinants (Grimaldi & Cricelli, 2019; Grimaldi et al., 2018).

[c] Determinants are arranged in alphabetical order, and refer to 2.1.2.4

[d] Dimension is defined as per Table 2.3.

[e] It includes the description, formula and variety.

[f] Rational refers to the purpose of the determinant and how it is interpreted.

[g] Some authors appear more than once because they use more than one determinant in the studies.

2.1.2.5 Composite indices

While there has been a number articles on patent proxies (Table 2.4), and patent value determinants (Table 2.5), there have been limited studies on composite indices of patent value. We identify patent value composite indices during our review of the articles, which combine one or more variables from Tables 2.4 and 2.5. We classify them as patent value proxies[1], consistent with our definition of patent value proxies. Table 2.6 provides an overview of the composite indices identified in alphabetical order, with their definition, how they have been used, and some examples. The majority of articles proposing composite indices are recent, i.e. published in the last decade, and concentrate on valuing patent portfolios rather than individual patents.

Table 2.6 Patent value composite indices

Composite Patent Index[a,b]	Value Assessment[c]	Level of Usage[d]	Authors (examples)[e]
Corporate market value, sales, return on equity	The index measures the market value, sales, and return on equity by combining the following determinants: HHI-index, current impact index, essential patent index	Portfolio	Chang (2012); Zhang et al. (2012)
Essential Technology Strength Index (ETS)	The index is constructed using the number of patents, the current impact index, and the essential patent index	Portfolio	Chen et al. (2007)
Patent Asset Index (PAI)	The index measures the strength of patent as a combination of competitive impact, market coverage, technology relevance, and portfolio size	Patent	Ernst & Omland (2011)
Patent Portfolio Value Index (PPVI)	The index is calculated using the technological scope, forward citation frequency, international scope, patenting strategy, and economic relevance	Portfolio	Grimaldi et al. (2015, 2018)
Patent Quality Index (PQI)	Patent quality is analysed using 4 determinants: number of claims, forward citations, number of backward citations, Family size	Patent	Lagrost et al. (2010); Lanjouw & Schankerman (2004); Squicciarini et al. (2013)
Potential Market of Patented Invention (PMPI)	The index is calculated by combining 5 elements: forward citations, grant decisions, number of families, renewals, and oppositions	Portfolio	van Zeebroeck (2011)

[a] The list is not exclusive, and provides the composite indices identified in the articles.

[b] We classify them as patent value proxies. A patent value proxy is a measure that can be used to approximate the value of a patented invention. It represents patent characteristics, which have been used mainly in the literature as dependent variables and are classified as ex-post indicators (van Zeebroeck & van Pottelsberghe de la Potterie, 2011a).

[c] This includes the composite index's definition, and its component variables from Tables 2.4 and 2.5.

[d] We identify the level of the analysis for these composite indices: portfolio level, or individual patent level

[e] When there is more than one study, the authors construct the composite index similarly.

Lanjouw & Schankerman (2004) propose a multiple proxy model which combines four patent determinants: number of claims, forward citations to the patent, backward citations in the patent application, and family size, similar to Squicciarini et al. (2013) who also use the patent family. Ernst & Omland (2011) develop a benchmarking index, which evaluates

[1] A patent value proxy is a measure that can be used to approximate the value of a patented invention. It represents patent characteristics, which have been used mainly in the literature as dependent variables and are classified as ex-post indicators (van Zeebroeck & van Pottelsberghe de la Potterie, 2011a).

a patent portfolio of a company in comparison with its competitors, and is based on the technological scope, market coverage, competitive impact, technology relevance and portfolio size. This is similar to Grimaldi et al. (2018), who propose a portfolio value index based on technological scope, forward citation frequency, international scope, patenting strategy and economic relevance (van Zeebroeck, 2011).

2.1.2.6 Methodological approaches deployed for patent valuation

We focus the methodological approaches currently deployed for patent valuation, identified in the review. Table 2.7 provides an overview of approaches, with examples. We review previous studies for the specific approaches and models used (2.1.2.6.1), the evaluation metrics (2.1.2.6.2), and the sample size (2.1.2.6.3).

2.1.2.6.1 Methodological approaches

The majority of articles (52%) use traditional regression methods for patent valuation using proxies and determinants, which use simply computed data, such as numeric and binary categoric types of data. These articles chose a dependent variable based on patent characteristics, i.e. forward citations, followed by the applicant's characteristics, i.e. firm market value. The most common approach used is the ordinary least square (OLS) regression[1], followed by Tobin Q regression[2]. Several scholars also make use of probit regression[3], where the dependent variable is binary categoric. Classifying observations based on their predicted probabilities is a type of linear binary classification model, which treats similar problems like the logistic regression.

The remaining articles fall within the approaches of artificial intelligence (AI) and other. About 34% of the articles are classified as other, using descriptive and conceptual methodologies to discuss the literature around the value of patents and technologies. From these, the majority are descriptive literature papers, which discuss one or more dimensions (Grimaldi et al., 2018), proxies (Aristodemou & Tietze, 2018b; Grimaldi & Cricelli, 2019), determinants (Lagrost et al., 2010) or composite indicators, followed by conceptual frameworks about the strategic nature of patent value (Ernst, 2003).

[1] Ordinary least square (OLS) regression is a type of linear least squares method for estimating the unknown parameters in a linear regression model. OLS chooses the parameters of a linear function of a set of explanatory variables by the principle of least squares, which minimises the sum of the squares of the differences between the observed dependent variable in the given dataset and those predicted by the linear function (James et al., 2013; Murphy, 2012).

[2] Tobin's Q, also known as Q ratio, is the ratio between a physical asset's market value and its replacement value (Tobin & Brainard, 1977).

[3] Probit regression estimates the probability that an observation with particular characteristics falls within a specific category (Agresti, 2015).

Table 2.7 Methodological approaches for assessing the value of patents

Methodological approach		No.[a]	Authors	Evaluation metrics[b]					Datapoints (000's)				Data Type[c]		Dep.Var [d]			
Main	Specific			Log-L	R^2	χ^2	Error	Other	<10	10-100	100-1000	>1000	Num.	Cat.	C	A	I	S
Artificial Intelligence	Artificial Neural Networks	2	Chen & Chang (2010d); Trappey et al. (2012)	-	-	-	2	-	2	-	-	-	2	1	1	1	-	-
	Decision Tree	2	Bass & Kurgan (2010); De Clercq et al. (2019)	-	-	-	1	-	-	1	-	-	1	-	1	-	-	-
	Naïve Bayes Classifier	1	Bass & Kurgan (2010)	-	-	-	1	-	-	-	1	-	1	-	1	-	-	-
	PageRank	1	Mariani et al. (2019)	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-
	Random Forest	2	Bass & Kurgan (2010); De Clercq et al. (2019)	-	-	-	1	-	-	1	-	-	1	-	1	-	-	-
Regression	Binomial (Negative, Linear Probability)	4	Alcácer & Gittelman (2006); Alcácer et al. (2009); Harhoff et al. (1999); Mariani & Romanelli (2007); Sapsalis et al. (2006)	3	2	1	-	1	3	1	1	-	3	2	3	-	-	1
	Heckman	1	Gambardella et al. (2007)	1	-	1	-	-	1	-	-	-	-	-	-	-	1	-
	Cox Proportional Survival Model	2	Lee et al. (2017)	-	-	1	-	1	1	-	-	-	-	-	1	-	-	-
	Logistic	2	Bass & Kurgan (2010); Verhoeven et al. (2016)	-	1	-	1	-	-	1	1	-	2	-	2	-	-	-
	Monte Carlo	1	Bessen (2008)	1	-	-	-	-	-	-	1	-	-	1	-	-	1	-
	Ordinalry Least Square (OLS)	13	Alcácer et al. (2009); Bessen (2009); Chen & Chang (2010d); Fischer & Leidinger (2014); Greenhalgh & Rogers (2006); Hall & MacGarvie (2010); Harhoff et al. (1999); Lanjouw & Schankerman (2004); Mowery et al. (1996); Sreekumaran Nair et al. (2012); Suzuki (2011); Tekic & Kukolj (2013)	3	11	-	-	1	8	2	3	-	10	7	2	7	2	-
	Panel	2	Frietsch et al. (2014); Neuhäusler & Frietsch (2013)	-	2	-	-	-	2	-	-	-	1	1	-	1	-	1
	Poisson	1	Messeni Petruzzelli et al. (2015)	1	-	-	-	-	1	-	-	-	-	1	1	-	-	-
	Probit	6	Gambardella et al. (2007); Harhoff & Reitzig (2004); Harhoff et al. (2003); Lanjouw & Schankerman (2001); Reitzig (2003, 2004)	4	4	4	-	-	5	1	-	-	-	5	-	1	5	-
	Tobit	2	Alcácer et al. (2009); Mowery et al. (1996)	1	-	-	-	1	-	-	2	-	-	1	2	-	-	-
	Tobin Q	6	Bakker (2017); Hall (2005); Hall & MacGarvie (2010); Lin et al. (2006)	1	2	1	-	-	1	1	4	-	4	3	4	2	-	-
Other	Conceptual Framework	8	Berg et al. (2018); Chen et al. (2007); Ernst (2003); Ernst et al. (2010); Ernst & Omland (2011); Grimaldi et al. (2018); Kogan et al. (2017); Lanjouw et al. (1998)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Descriptive Articles	10	Aristodemou & Tietze (2018b); Breitzman & Mogee (2002); De Saint-Georges & Van Pottelsberghe De La Potterie (2013); Dolfsma (2011); Grimaldi et al. (2015); Grupp & Mogee (2004); Kalutkiewicz & Ehman (2014); Lagrost et al. (2010); Narin & Hamilton (1996); Striukova (2007)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Kaiser-Meyer-Olkin Factor (KMO) Analysis	1	Hagedoorn & Cloodt (2003)	-	-	-	-	1	-	1	-	-	-	-	-	-	-	-
	Structural Equation Modelling	4	Deng (2007); Lanjouw & Schankerman (2004); Ma et al. (2019); Suzuki (2011)	-	2	-	-	-	1	-	2	-	-	-	-	3	-	1
	Survey (Block Design, Delphi)	2	Albert et al. (1991)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Systematic Review	1	Adams (2006)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			Total	15	24	8	6	5	26	9	15	1	25	22	20	15	9	3

[a] The total number of articles is not the same as the identified articles in 2.1.1.1 because some articles deploy more than one methodology.

[b] Evaluation metrics have been grouped together: R^2 includes variations such as McFaddens/ pseudo/ adjusted R^2, χ^2 includes variations such as Wald χ^2, other includes F-test, % variance test, z-score.

[c] Types of data used in the studies are either numeric or binary-categoric

[d] The dependent variable categories are defined after Table 2.2a and van Zeebroeck & van Pottelsberghe de la Potterie (2011b). *C* represents patents characteristics, *A* represents applicants information, *I* represents inventor information, and *S* represents the timing and filing strategy of the invention

2.1.2.6.2 Evaluation metrics

The articles use a variety of evaluation metrics, to evaluate the suitability of the proposed models to measure patent value. The regression approaches focus on the use of R^2 and its variations. Particularly, the OLS regression and probit regression approaches seem to focus on the use of R^2 together with the $Log-L$ measure. The remaining of the regression studies use measures such as χ^2 and its variations. This is in contrast to AI approaches, which entirely focus on the error evaluation metric, which shows the difference between predicted and actual output.

2.1.2.6.3 Sample size

The sample size varies according to the different approaches deployed. The majority of the regression studies have less than 10000 datapoints, using mainly numeric, and binary categoric variables. Only 1 study has a sample size greater than 1000000, found in the AI approaches. There are a few studies that have used datapoints in the range of 100000-1000000, which are mostly comprised by numeric types of data in linear models, with a range of R^2 values between 0.3-0.7. In addition, the majority of the dependent variables in the regression methodologies are concentrated around patent characteristics, followed by applicant characteristics.

2.2 Intellectual Property Intelligence (IPI)

2.2.1 Background

Increased data availability presents an opportunity for better decision making, policy and strategy development, to introduce the next generation of innovative technologies (Günther et al., 2017). We define *Intellectual Property Intelligence (IPI)* as the data science of analysing large amount of intellectual property information, specifically patent data, with artificial intelligence (AI) methodologies, such as machine learning and deep learning, to discover relationships, trends and patterns in the data for decision making[1]. Data as such presents value for enabling a competitive data-driven economy (OECD, 2017, 2019a,b). This definition and the associated AI-based approaches form the basis of the emerging field of IPI research.

It is important to understand the process of analysing patent data. Trippe (2015) has produced a WIPO guide, which explains a large number of concepts on patent analysis[2].

[1] This definition is an advancement to the one proposed by Aristodemou & Tietze (2018b) for *Intellectual Property Analytics (IPA)*, to focus on artificial intelligence (AI) methodologies.

[2] Oldham & Fried (2016), URL: https://wipo-analytics.github.io/

With the recent advancements of AI, there has been a positive amount of activity around the different methodologies involved that could be applied to patent data (Aristodemou & Tietze, 2017b; Lupu, 2018; Trappey et al., 2020a; WIPO, 2019b). Moehrle et al. (2010) argue that the analysis of patent data in a business context consists of three stages: the pre-processing stage, the processing stage and the post processing stage. This is similar to Abbas et al. (2014), who present a generic patent analysis workflow, with specific purpose analysis types. Raturi et al. (2010) argues that this process is a complementary process to the innovation cycle. Similarly, Baglieri & Cesaroni (2013) argue that patent analysis is a form of patent intelligence to support decision making (Bonino et al., 2010).

With the rise of AI and increased computational resources, there has been an increase in the usage of methodological approaches such as machine learning (ML) and deep learning (DL), previously not been deployed, to analyse patent data (Aristodemou & Tietze, 2018b). Given the rise of interest for AI in the analysis of patent data (OECD, 2019a; WIPO, 2019b), we review the methodological approaches deployed for the analysis of patent data. More specifically, we synthesize a taxonomy for these (2.2.2.1), focusing on artificial neural network (ANN) methodologies and the architectures deployed (2.2.2.2). We then explore the areas of application of IPI, concentrating on the AI methodologies deployed for patent value (2.2.2.3).

2.2.1.1 Review Methodology

We review the IPI literature, with the aim to synthesize a taxonomy of AI methodologies analysing patent data, and identify the relevant ones deployed for patent value. To carry out the review[1], the narrative and scoping literature review approaches have been adopted (Cronin et al., 2008; Paré et al., 2015), and a search strategy[2] has been developed (Robson, 2011). Fig.2.3 shows the process flow for the review. The articles on IPI are identified from the Scopus database (Falagas et al., 2008). Focusing on recent literature, the search is constrained to articles published after 2000, to the fields of business, computer science, engineering, social science and mathematics. The core review identifies 57 articles. For these, we review in detail the architectures and approaches deployed for the analysis of patent data with AI methodologies, with a focus on the ones deployed for patent value.

[1]The review draws predominantly on on-line resources (articles), and is supported by the narrative perspective using book references. This inherently introduces a gap in the literature review. We aim to reduce that gap using book and report references to complement our arguments and explanations.

[2]We search within the title, abstract and key words for various terms such as 'patent', 'patent data', 'patent analysis' and 'intellectual property data'. The search is then narrowed to documents that also contain either in the title or the abstract or in the key words, the terms 'machine learning', 'machine learning models', 'neural networks', 'deep learning' and 'artificial intelligence'. The search is effective on 08.01.2018.

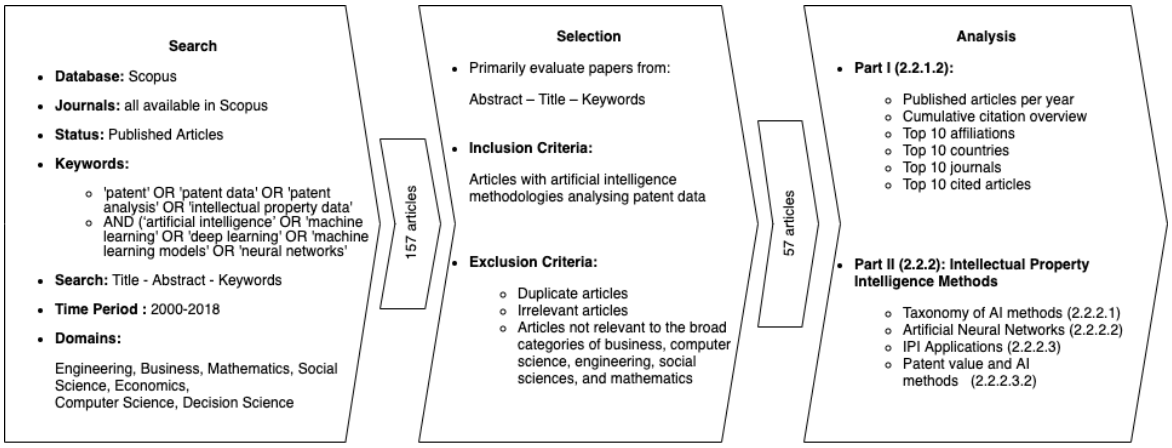


Fig. 2.3 Process flow of the review on Intellectual Property Intelligence

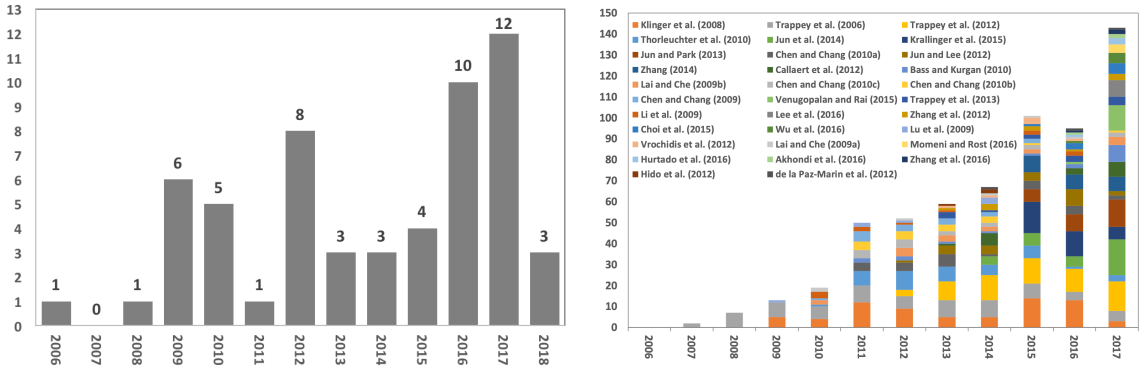
2.2.1.2 Bibliographic analysis results

The first level of the analysis focuses on the bibliographic information of the 57 articles (n_1). The number of articles have increased since 2009, reaching a peak of 12, indicating an increasing interest in the field, together with an increase in the citations to 153. Fig.2.4 shows the number of papers per year since the year 2000 (Fig.2.4a), and the cumulative citation overview per article per year (Fig.2.4b).

Table 2.8 shows the top 10 affiliations, countries, journals and citations of the 57 articles. Asia is the leading continent, which is also supported by the fact that it accounts for 47% of the world's patent application filings (WIPO, 2019b). While a late comer in the patent field, it is forerunner in the applications of AI for patent data (OECD, 2019a). This is supported by Table 2.8b, with the top 3 countries are Taiwan, South Korea and China respectively. European countries also show strong influence in this domain, with 19% of the share total. Moreover, the top two journals, which account for 30% of the articles (15% each) are Technological Forecasting and Social Change and Scientrometrics. The top 10 journals account for 33 articles, indicating that articles in this field are fragmented in 34 journals.

The most cited article is Klinger et al. (2008) with 70 citations, followed by Trappey et al. (2006) with 68 citations, and Trappey et al. (2012) with 61 citations. However, the article with the highest citation frequency[1] is Krallinger et al. (2015) with 11.00, followed by Trappey et al. (2012) with 10.17, and Klinger et al. (2008) with 7.00. 8 out of the 10 articles are published since 2010, which indicates an increase in the application of AI methodologies with the analysis of patent data. Together with Table 2.8a, where all the top 10 affiliations are in Asia, this indicates the importance and progression Asia has made in this field.

[1]Citation frequency is defined as the total number of citations over the age of the article.



(a) (b)

Fig. 2.4 Plots of: 2.4a number of articles published per year (n_1=57) since 2000; 2.4b cumulatitve citation overview per article per year, for articles with > 2 citations

Table 2.8 Top 10 affiliatons (2.8a), countries (2.8b), journals (2.8c), and cited articles (2.8d), for articles analysing patent data with Artificial Intelligence Methodologies (n_1=57 articles)

(a) Affiliations (n_2=128 observations)

Affiliation[a]	No. of obs.	Share %
National Tsingua University, CN	7	5%
National Chiao Tung University, TW	6	5%
Korea University, KR	5	4%
Cheongju University, KR	5	4%
National Yunlin University, TW	5	4%
University of Nis, RS	4	3%
Korea Institute of S & T., KR	3	2%
Gainia Intellectual Asset Services, TW	2	2%
Chung Hua University, TW	2	2%
Beijing Institute of Technology, CN	2	2%
Total	41	33%

[a]Articles with one or more affiliations are multi-counted.

(b) Countries (n_2=71 observations)

Country[a]	No. of obs.	Share %
Taiwan	18	25%
South Korea	12	17%
China	8	11%
United States	6	8%
Germany	4	6%
Serbia	4	6%
Spain	3	4%
Belgium	2	3%
Japan	2	3%
Hong Kong	1	1%
Total	60	84%

[a]Articles with one or more countries are multi-counted.

(c) Journals

Journal[a]	No.	Share %
Technological Forecasting And Social Change	8	14%
Scientometrics	8	14%
Expert Systems With Applications	4	7%
World Patent Information	3	5%
Sustainability Switzerland	3	5%
Database The Journal Of Biological Databases	2	4%
International Journal Of Applied Engineering Research	2	4%
Physica A Statistical Mechanics And Its Applications	2	4%
Advanced Engineering Informatics	1	2%
Applied Soft Computing Journal	1	2%
Total	33	61%

[a]The 57 articles are published in 34 journals.

(d) Cited Articles

Article[a]	Citations	Cit. Freq.
Klinger et al. (2008)	70	7.00
Trappey et al. (2006)	68	5.67
Trappey et al. (2012)	61	10.17
Thorleuchter et al. (2010)	39	4.88
Jun et al. (2014)	34	8.50
Krallinger et al. (2015)	33	11.00
Jun (2013)	27	5.40
Chen & Chang (2010b)	25	3.13
Jun & Lee (2012)	23	3.83
Zhang (2014)	22	5.50

[a]Citations frequency: total number of citations over the age of the article.

2.2.2 Intellectual property intelligence methods

Several analytical methodologies have been used with patent data (Abbas et al., 2014; Trippe, 2015). Specifically, Aristodemou & Tietze (2018b) focus on the artificial intelligence (AI) methodologies deployed with the analysis of intellectual property (IP) data. We review the literature, to identify and understand the AI methodologies analysing patent data, also defined as intellectual property intelligence (IPI). Firstly, we construct an advanced taxonomy of AI methodologies used with patent data, to identify the learning paradigms, types of application, methodologies and algorithms delpoyed (2.2.2.1)[1,2].

From the taxonomy, we identify the limited application of deep learning (DL)[3]. We then review the articles on artificial neural networks (ANN) (2.2.2.2), and focus on the ones, which deploy the multi-layer perceptron (MLP) network architecture[4]. We aim to identify the current state of the art in the deep learning methodologies deployed for the analysis of patent data. Finally, we focus on the areas of application of IPI, and specifically on the articles that focus on patent value (2.2.2.3), to comprehend the methodologies and models developed by previous scholars.

2.2.2.1 Taxonomy of artificial intelligence methodologies for patent data analysis

Table 2.9 presents the taxonomy of the artificial intelligence (AI) methodologies[5] deployed to analyse patent data (Barredo Arrieta et al., 2020). We identify 10 collective AI methodological

[1]Learning paradigms represent the way and purpose of learning. In artificial intelligence (AI), there are 3 learning paradigms: supervised, unsupervised, and reinforcement. The majority of AI methodologies deployed today are hybrid, i.e. involve a combination of the above. Supervised learning is when a learning task infers a function from the analysis of the training data, given a set of mapped input-output pairs, and can determine the mapping of new examples (Bishop, 2006; Goodfellow et al., 2016). Unsupervised learning is often used to cluster, associate or summarise data to reveal information or underlying relatioships (Goodfellow et al., 2016). Reinforcement learning is when a learning task carries a punishment and reward with every inference from an input (Murphy, 2012).

[2]The learning paradigms have application types associated with them. Supervised learning can be applied in a regression or classification problem task. Unsupervised learning can be applied in a classification, clustering or dimensionality reduction problem task. Reinforcement learning can be adapted to the specific problem task.

[3]Deep Learning (DL) has many different learning paradigms, types of application, methodologies and algorithms. For the purpose of this research, we use *Deep Learning (DL)* as artificial neural networks (ANN), in supervised learning paradigms, defined by the depth of the credit assignment paths, which are chains of possibly learnable, causal links between inputs and outputs, i.e. finding weights that make the neural network exhibit desired behaviour (Schmidhuber, 2015). These are also known as deep (and wide) neural networks (Cheng et al., 2017; Goodfellow et al., 2016; Shaked et al., 2016).

[4]Multi-layer perceptron (MLP) network architectures primarily form the basis of deep learning models (Schmidhuber, 2015), and is the most widely researched and implemented methodology in other fields (Basheer & Hajmeer, 2000; Bishop, 2006).

[5]For the purpose of simplicity, we refer the readers to the literature by Bishop (2006); Goodfellow et al. (2016); Murphy (2012); Schmidhuber (2015), for an introduction and in-depth understanding of the artificial intelligence methodologies and respective algorithms.

approaches, which are split further into 32 algorithms.

Table 2.9 Artificial intelligence methodologies deployed to analyse patent data

Paradigm[a,b]	Type[c]	Methodology[d,e]	Algorithm	Authors (examples)[f]
Supervised learning	Classification/ Regression	Artificial Neural Networks (ANN)	Backpropagation (BP)	Chen & Chang (2009, 2010a,b,c); Chiang et al. (2011); Jokanović et al. (2017); Kim & Lee (2017); Kyebambe et al. (2017); Lai & Che (2009a); Lee et al. (2018); Riedl et al. (2016); Trappey et al. (2012, 2006, 2013); Venugopalan & Rai (2015); Zhang et al. (2012)
			Evolutionary Algorithm	de la Paz-Marín et al. (2012)
			Extension Theory Learning	Lai & Che (2009b)
			Extreme Learning Machine (ELM)	Jokanović et al. (2017); Marković et al. (2017)
			Fuzzy Interference	Marković (2017)
			Growing Cell Structure	Sung et al. (2017)
		Decision tree	CART	Choi et al. (2015); Zhu et al. (2015)
			C4.5	Bass & Kurgan (2010)
		Deep Belief Networks (DBN)	Backpropagation (BP)	Lee et al. (2017)
		Ensemble	Bootstrapping	Klinger et al. (2008)
			Random Forest	Bass & Kurgan (2010)
			Stacking	Leaman et al. (2016)
		Regression	Linear Regression	Jun (2013); Jun & Lee (2012); Jun et al. (2014); Lai & Che (2009b)
			Logistic Regression	Bass & Kurgan (2010); Han et al. (2017); Hido et al. (2012)
		Statistical and probabilistic modelling	Conditional random fields (CRF)	Akhondi et al. (2016); Klinger et al. (2008); Krallinger et al. (2015); Zhang et al. (2016b)
			Hidden Markov Model (HMM)	Lee et al. (2016)
			Latent Dirichlet Allocation (LDA)	Govindarajan et al. (2018); Suominen et al. (2017)
			Naive Bayes	Bass & Kurgan (2010); Zhu et al. (2015)
		Support Vector Networks (SVN)	Support Vector Clustering (SVC)	Jun et al. (2014)
			Support Vector Machine (SVM)	Krallinger et al. (2015); Kyebambe et al. (2017); Li et al. (2009); Lu et al. (2009); Vrochidis et al. (2012); Wu et al. (2016); Zhang et al. (2009); Zhang (2014); Zhang et al. (2016b)
Unsupervised learning	Clustering	Clustering	Girvan-Newman (GN)	Sung et al. (2017)
			K-means Algorithm (and derivations)	Jun (2013); Jun et al. (2014); Trappey et al. (2017b)
			Self Organising Maps (SOM)	Chen & Chang (2009, 2010a,b); Wu et al. (2016)
	Dimensionality reduction	Pre-processing data	Linear Discriminant Analysis (LDA)	Callaert et al. (2012); Venugopalan & Rai (2015)
			Multi-Dimensional Scaling (MDS)	Lamirel et al. (2003)
			Principal Component Analysis (PCA)	Jun et al. (2014); Lai & Che (2009b); Trappey et al. (2012)
			Quadratic Discriminant Analysis (QDA)	Venugopalan & Rai (2015)
			Singular Value Decomposition (SVD)	Jun et al. (2014)
		Text mining	Dictionary-based	Akhondi et al. (2016); Krallinger et al. (2015)
			Natural Language Processing (NLP)	Han et al. (2017); Krallinger et al. (2015)
			Rule-based	Bass & Kurgan (2010); Krallinger et al. (2015)
			Semantic-based	Trappey et al. (2013); Wu et al. (2012b); Zhang et al. (2016b)
Reinforcement Learning	Reward/ penalty	Reinforcement Learning	SARSA	Tenorio-González & Morales (2018)

[a] We base the synthesis of the taxonomy on the WIPO report on AI Technology Trends (WIPO, 2019b), and the theory underpinning artificial intelligence (AI), machine learning (ML) and deep learning (DL) (Goodfellow et al., 2016; Murphy, 2012). We identify the learning paradigm and type of application of the AI methodologies deployed, and the specific algorithms, giving some examples from the literature. WIPO (2019b), URL: https://www.wipo.int/tech_trends/en/artificial_intelligence/.

[b] The table is structured according to the learning paradigms, and then arranged in alphabetical order according to the methodology deployed, and algorithm.

[c] Within the learning paradigm, the type indicates the type of model build for the application.

[d] It's important to note that this is not an absolute taxonomy since several algorithms can have more than one learning paradigm or types of application. The list is not exhaustive.

[e] We refer the readers to the literature by Bishop (2006); Goodfellow et al. (2016); Murphy (2012); Schmidhuber (2015), for an introduction and in-depth understanding of the artificial intelligence methodologies and respective algorithms.

[f] Several studies can be classified in more than one methodology.

The majority of articles are clustered around supervised and unsupervised paradigms of learning, with only one article using reinforcement learning. This is reflective of the artificial intelligence field development, with reinforcement learning becoming recently popular (WIPO, 2019b).

Supervised learning has the highest share of articles, similar to the trend from other fields (Ozturk et al., 2020; Tietze et al., 2020b). With artificial neural networks (ANN) being the most popular methodology deployed, the backpropagation algorithm (BP) is the most used algorithm, and remains until today the central algorithm for the development of networks. This is because ANNs are versatile, robust, scalable and can handle high dimensionality tasks (Bengio et al., 2007; Bishop, 2006). In addition, the type of application models concentrate on classification and regression for supervised learning. For unsupervised learning, there is a large number of articles around dimensionality reduction[1]. This is expected since these methodologies have the ability to summarise data with a large number of dimensions (Goodfellow et al., 2016), which is an essential step for the analysis involving patent data (Moehrle et al., 2010)[2].

2.2.2.2 Artificial Neural Networks

Artificial neural networks (ANNs) are computational methodologies that can solve many complex real-world problems (Basheer & Hajmeer, 2000). They have been proposed in the 1940s, but recently have gained remarkable attention for patent data due to the capacity of the information they can process, as the computer science has evolved (Hagan et al., 1995). ANNs are modelled after biological neurons, with complex functions (Gupta, 2000; Murphy, 2012). They tend to outperform traditional methods (such as regressions) when the dimensionality and non-linearity of the problem increases, since they have a high noise tolerance, earning and generalisation capabilities (Basheer & Hajmeer, 2000; Hill et al., 1993; Lee et al., 1989; Sargent, 2001).

Several ANN methodologies have been developed over the years, with many algorithms, network architectures, parameter and network optimisation techniques (Hudson & Postma, 1982; Maren, 1991; Murphy, 2012; Schmidhuber, 2015). The most widely researched and implemented methodology in other fields is the multi-layer perceptron (MLP)[3,4,5],

[1] Golstein (2018), URL: https://www.sharper.ai/taxonomy-ai/.

[2] Brownlee (2019a), URL: https://machinelearningmastery.com/types-of-learning-in-machine-learning/.

[3] The artificial neural network (ANN) approach consists of fixing a number of basis functions, which are adaptive, i.e. parametric forms of these basis functions are used so their parameters are adapted (Bishop, 2006).

[4] Van Veen (2016), URL: https://www.asimovinstitute.org/neural-network-zoo/

[5] The multi-layer perceptron (MLP) is a feed-forward artificial neural network (ANN). The MLP is a series of logistic regressions stacked on top of each other (continuous non-linearities), with the final layer being either another logistic regression, a linear or non-linear regression model, depending on the problem under

which forms the basis of deep learning (Basheer & Hajmeer, 2000; Bishop, 2006), with the backpropagation (BP) algorithm still remaining the central learning algorithm (Schmidhuber, 2015). We review the articles that deploy the ANN MLP network architectures to identify the different methodological characteristics for the analysis of patent data, focusing on the output variables and sample size the scholars use (2.2.2.2.1), the parameter and network optimisation of the models (2.2.2.2.2), and the evaluation measures used to evaluate the models (2.2.2.2.3).

Table 2.10 shows the articles, that deploy the ANN MLP network architecture with the BP algorithm on patent data. There are 13 articles in total, which shows how limited the field still is. The majority of articles (85%) have been published after 2010, indicating the interest in these ANNs for patent analysis is growing. Reviewing the articles' aim, 38% of them belong to the topic of economic valuation (i.e. corporate market value, GDP), 31% on forecasting (i.e. emerging technologies, multi-technology convergence), and 31% on document classification (Aristodemou & Tietze, 2018b). There is limited association of these directly to patent value or forecasting of patent value. Moreover, 54% of the articles are classification applications, with the remaining being regression applications.

2.2.2.2.1 Output variables and sample size

We observe that 31% of the articles use the IPC classification of a patent as the output variable (Table 2.10)[1]. In addition, 38% of the articles have some association to value, with output variables such as firm market value, GDP or forward citations (Kim & Lee, 2017; Lee et al., 2018). Moreover, 54% of the articles use a patent characteristic as a proxy output variable, such forward citations or IPC classification, with only one article using a composite index in the form of trading quality (Trappey et al., 2012). All the articles in Table 2.10 have a small sample size, with only one above 10,000 datapoints, one above 1,000, and the rest below.

2.2.2.2.2 Parameter and network optimisation

We observe that almost all the applications of ANNs are shallow neural networks since all the models, except one (Kim & Lee, 2017), have one hidden layer (Mhaskar et al., 2017; Mhaskar & Poggio, 2016)[2]. This suggests that there is limited application of deep learning (Schmidhuber, 2015). In addition, the majority of these models require a high number of

investigation (Murphy, 2012; Pal & Mitra, 1992).

[1] An output variable is a variable being predicted in supervised learning. In statistics, this is also known as the dependent variable, or target variable.

[2] A shallow neural network is a term used to describe neural networks with one hidden layer. We adopt the terminology by Murphy (2012), where a 2 layer ANN is a network with 2 layers of adaptive weights, i.e. 1 hidden layer (Mhaskar & Poggio, 2016).

Table 2.10 Multi-layer perceptons (MLP) feed-forward artificial neural network (ANN) architectures

Author[a,b]	Type[c]	Output variable[d]	Sample size[e]		Parameter optimisation[f]		Layer activation function		Network Optimisation[g]			Evaluation metrics[h]						
			Train	Test	Hidden layer neurons	Layers[i]	Hidden	Output	Learning rate	Momentum	Epochs	MAE	MAPE	RMSE	Accuracy	Precision	Recall	F1-score
Lee et al. (2018)	Classification	Forward citations	28286	7071	4	2	Sigmoid	Sigmoid	-	-	1000	-	-	-	0.910	0.773	0.373	-
															0.837	0.510	0.343	
															0.711	0.448	0.370	
Jokanović et al. (2017)	Regression	GDP	-	-	6	2	Sigmoid	-	-	-	1000	-	-	0.040	-	-	-	-
Kim & Lee (2017)	Regression	Forward citations	-	-	50	3	Tangent	-	0.4	0.9	10000	-	18.33	-	-	-	-	-
													21.08					
													12.31					
Kyebambe et al. (2017)	Classification	Emerging Technology	-	-	-	-	-	-	-	-	-	-	-	-	-	0.540	0.540	0.540
Venugopalan & Rai (2015)	Classification	IPC	4298	1842	4	2	Sigmoid	-	-	-	-	-	-	-	0.813	-	-	-
Trappey et al. (2013)	Classification	IPC	233	110	-	-	Sigmoid	-	-	-	-	-	-	-	-	0.909	0.930	0.926
Trappey et al. (2012)	Classification	Trading quality	260	101	-	-	-	-	0.2	0.8	10000	0.290	-	-	0.850	-	-	-
Zhang et al. (2012)	Regression	Sales	462	116	1	2	Sigmoid	Linear	0.9	0.8	100000	0.016	0.053	0.027	-	-	-	-
Chiang et al. (2011)	Classification	IPC	120	50	26	2	Sigmoid	Sigmoid	0.2	0.8	1000	-	-	0.135	0.830	-	-	-
Chen & Chang (2009, 2010a,b)[j]	Regression	Firm market value	272	70	3	2	Sigmoid	Linear	-	-	100000	0.027	0.045	0.051	-	-	-	-
Trappey et al. (2006)	Classification	IPC	300	124	-	2	Tangent	Sigmoid	-	-	80	-	-	-	-	0.919	-	-

[a]The table is arranged in descending order, i.e. the most recent article is found at the top.

[b]The table is a subset of Table 2.9, and focuses on the methodological characteristics of multi-linear perceptrons (MLP) artificial neural networks (ANN) for the analysis of patent data. Empty cells correspond to non-existing values in the literature. The list of articles is not exhaustive.

[c]Within the learning paradigm, the type indicates the type of model build for the application.

[d]An output variable is a variable being predicted in supervised learning. In statistics, this is also known as the dependent variable, or target variable.

[e]Sample size refers to the split and number of datapoints between the training set and the testing set for the algorithms.

[f]Parameter optimisation refers to the hyperparameters for neural networks. This is explained in Chapter 4 and for simplicity we report the most common hyperparameters.

[g]Network optimisation refers to the hyperparameters relating to the network architecture and the trainable weight parameters. This is explained in Chapter 4 and for simplicity we report the most common parameters.

[h]Evaluation metrics evaluate the model's generalising ability. Regression evaluation metrics include mean absolute error (MAE), mean absolute percentage error (MAPE), root mean square error (RMSE). Classification evaluation measures include accuracy, precision, recall, and F1-score (see 4.3). For simplicity, not all evaluation metrics are reported.

[i]We adopt the terminology by Murphy (2012), where a 2 layer ANN is a netwrok with 2 layers of adaptive weights, i.e. 1 hidden layer.

[j]These 3 articles are grouped together due to their high similarity.

epochs to be trained, with relatively high learning rates. This suggests that either ANN MLP models based on patent data require more time to converge, or the architecture network parameters are partly suitable to represent the problem the articles are tackling.

2.2.2.2.3 Evaluation metrics

The suitability of evaluation metrics is driven by the application type. The majority of articles focus on classification types of applications (54%), with the remaining to be regression types[1]. Some patent scholars report low mean absolute error (MAE), which is partly driven by the model complexity, i.e. the low number of input layer nodes and hidden layer nodes (Chen & Chang, 2009, 2010a,b), where others report a relatively large MAPE (Kim & Lee, 2017). These variations in the metrics are also driven by the scale of input data together with the model complexity. Scholars that focus on classification problems, report relatively high accuracy values when predicting the likely IPC classification of a patent (Trappey et al., 2012, 2006). This is partly because of two reasons: firstly, the model complexity is fairly small, with a small sample size, and secondly, some scholars restrict their sample size to a specific area, and thus that model is not able to universally generalise. However, there are some studies that report relatively low values of accuracy when forecasting forward citations over a time horizon, partly driven by the low recall values (Lee et al., 2018).

2.2.2.3 Intellectual property intelligence applications

Intellectual property intelligence, and artificial intelligence (AI) methodologies have been applied in a variety of areas within innovation management (Aristodemou & Tietze, 2018b). We expand the list of articles identified in 2.2.1.1 and Table 2.9 using the narrative approach (Cronin et al., 2008) to include recently published articles and working articles[2], between 2018-2020. We review the articles, and based on the aim of the articles we synthesize them into 3 areas of application. The areas are: knowledge and technology management, information retrieval and management, and economic development and valuation[3,4]. We

[1] Evaluation metrics such as mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE) are suitable for regression problems, where as accuracy, precision, recall, and F1-score are suitable for classification problems.

[2] In the computer science field, working paper articles are a way to publish research fast, while waiting for journal reviews. This shows the research area progression and are available almost immediately in archive.

[3] It is important to note that these areas of application are not mutually exclusive, and one article might belong to one or more areas of application. For simplicity, the most prominent theme within the article is identified for the article classification.

[4] In this research, we focus on the valuation of patents with artificial intelligence (AI) methodologies. We refer the readers to the article by Aristodemou & Tietze (2018b) for a description and in-depth discussion of the articles and the AI methodologies deployed in the areas of application presented in Table 2.11.

provide an overview of these in 2.2.2.3.1, and focus on the valuation of patents with AI methods in 2.2.2.3.2.

2.2.2.3.1 Areas of application of artificial intelligence approaches with patent data

Table 2.11 shows the areas of application of AI methodologies with patent data. The majority of articles are knowledge and technology management, followed by information retrieval, and economic development and valuation. The number of articles after 2010 is higher than before 2010, and between 2018-2020 there is an increase[1]. For the areas of knowledge and technology management and information retrieval, the number of articles published after 2010 is higher because the focus was on traditional methods (Lupu et al., 2011), where as for economic development the spread is equal. The shift between 2010 and after 2010 can be partly explained by the increase in computational resources and the rise of AI and Big Data (Aristodemou & Tietze, 2017b).

Table 2.11 Application areas of artificial intelligence (AI) methodologies with patent data

Area[a]	Authors (examples)	No. < 2010[b,c]	No. 2011-2018	No. 2018-2020
Economic Development and Valuation	Bass & Kurgan (2010); Chen & Chang (2009, 2010a,b,c,d); Choi et al. (2020); Jokanović et al. (2017); Karanikić et al. (2017); Lai & Che (2009a,b); Lee et al. (2016, 2018, 2017); Mariani et al. (2019); Marković (2017); Noh & Lee (2020); Trappey et al. (2019); Woo et al. (2019); Zhang et al. (2012)	8	6	6
Information Retrieval and Management of Information	Abdelgawad et al. (2020); Abood & Feltenberger (2018); Akhondi et al. (2016); Callaert et al. (2012); Choi et al. (2019); Govindarajan et al. (2019b); Hu et al. (2018b); Klinger et al. (2008); Krallinger et al. (2015); Leaman et al. (2016); Li et al. (2009); Lu et al. (2020); Riedl et al. (2016); Trappey et al. (2017a, 2020b); Venugopalan & Rai (2015); Vrochidis et al. (2012); Wu et al. (2012a); Zhang et al. (2018); Zhang (2014); Zhang et al. (2016b); Zhu et al. (2015)	2	12	10
Knowledge Management and Technology Management	Chiang et al. (2011); Choi et al. (2015); de la Paz-Marín et al. (2012); Govindarajan et al. (2018, 2019a); Han et al. (2017); Helmers et al. (2019); Hido et al. (2012); Jun (2013); Jun & Lee (2012); Jun et al. (2014); Kim & Lee (2017); Kyebambe et al. (2017); Lamirel et al. (2003); Lee et al. (2016, 2017); Lu et al. (2009); Momeni & Rost (2016); Seo et al. (2019); Sung et al. (2017); Tenorio-González & Morales (2018); Thorleuchter et al. (2010); Trappey et al. (2012, 2006, 2017b, 2013); Uhm et al. (2020); Wu et al. (2016); Wu (2019); Zhai et al. (2019); Zhang et al. (2009)	5	18	8
Total		15	36	24

[a]The table includes articles that are similar but published in different journals from one or more of the same authors. The list of articles include the ones identified in Table 2.9, and additional working paper articles between 2018-2020 (see 2.2.2.3). The table is arranged in alphabetical order according to the area column.

[b]We look at the distribution of articles before 2010 (including 2010), between 2011-2018 (excluding 2018), and between 2018-2020.

[c]The year 2010 is indicative of when popularity around artificial intelligence (AI) methodologies has increased due to advancements in computer science (Murphy, 2012; Schmidhuber, 2015).

2.2.2.3.2 Patent value with artificial intelligence methodologies

Focusing on articles in the area of economic development and valuation, and in particular patent valuation, we concentrate on the most relevant articles[2]. While there are 20 articles

[1]This can also be attributed to the inclusion of working paper articles.

[2]With the focus on the analysis of patent data (see 1.1.2 and 2.1.2), we can also define *intelligent patent value (IPV)* as the value of a patent arising or being represented by patent data determinants and proxies, i.e. arising from its technological significance (value of the invention) with the potential to develop a market (patent

in total in the area of economic development and valuation (Table 2.11), only 8 are directly relevant of deploying AI methodologies for the value of patents, with 3 of these published before 2018, and 5 published from 2018 onwards[1]. Table 2.12[2] shows the 8 relevant papers for deploying AI methods to value patents. The 8 articles are reviewed according to the approach they follow (2.2.2.3.2.1), the sample size and the data type (2.2.2.3.2.2), the parameter optimisation (2.2.2.3.2.3) and the evaluation metrics used (2.2.2.3.2.4).

Choi et al. (2020) propose an approach to evaluate the business potential of patents. They apply an ANN with BP to predict the likelihood that a patent will be renewed until its maximum expiration date. Trappey et al. (2019) deploy an ANN with the BP to classify the value of patents, defined as being a standard essential patent (SEP), within the IoT industry. Noh & Lee (2020) focus on forecasting forward citations, as a proxy for technological impact, using an ANN in the telecommunications area. Similarly, Lee et al. (2018) propose an ANN approach to identify emerging technologies using forward citations. Woo et al. (2019) propose a k-nearest neighbour (k-NN) clustering algorithm to screen early stage ideas. Trappey et al. (2012) develop a screening ANN model to improve patent quality, with a high quality patent defined as having been sold or licensed.

2.2.2.3.2.1 Methodological approach

Almost all the 8 articles follow the supervised learning paradigm, using ANN with the BP algorithm apart from Woo et al. (2019), which use the k-NN algorithm. Also, 63% of the articles are classification applications and the rest are regressions. The majority of articles (63%) use patent characteristics as output variables, with only Trappey et al. (2019), Trappey et al. (2012), and Chen & Chang (2010c) focusing on SEP, trading quality, and firm market value respectively (Chen & Chang, 2009). The most used output variable is forward citations with 3 studies (Lee et al., 2018; Noh & Lee, 2020; Woo et al., 2019), with only 1 recent study using renewals (Choi et al., 2020), and generality (Woo et al., 2019).

2.2.2.3.2.2 Sample size and data type

The majority of the studies have small datasets, constrained to a specific technological area. The study by Choi et al. (2020) has a larger dataset of 200000 datapoints, followed by 3 studies higher than 10000 datapoints (Lee et al., 2018; Noh & Lee, 2020; Woo et al., 2019). Moreover, all models use fairly simplistic numeric and binary categoric features, with only

premium) on dimensions such as economic, technological, social and strategic (see 2.1.2.2), and which is measured or analysed by artificial intelligence (AI) and data science methodologies (Aristodemou & Tietze, 2018b).

[1] Of these 5 articles, 3 have been published since November 2019 (Choi et al., 2020; Noh & Lee, 2020; Trappey et al., 2019).

[2] Table 2.12 is a subset of Table 2.11 and is synthesized from Tables 2.7 and 2.10.

Woo et al. (2019) making use of an abstract-keyword categoric matrix to utilise some text. All the other studies use limited features in the range of 5-24.

Table 2.12 Articles deploying artificial intelligence (AI) methods for valuation purposes

Author[a,b,c]		Choi et al. (2020)	Noh & Lee (2020)	Trappey et al. (2019)	Lee et al. (2018)	Woo et al. (2019)	Trappey et al. (2012)	Chen & Chang (2009, 2010c)
Approach	Paradigm[d]	Supervised	Supervised	Supervised	Supervised	Unsupervised	Supervised	Supervised
	Type[e]	Classification	Regression	Classification	Classification	Classification	Classification	Regression
	Methodology[f]	ANN	ANN	ANN	ANN	Clustering	ANN	ANN
	Algorithm[g]	BP	BP	BP	BP	k-NN	BP	BP
	Output variable[h]	Renewals	Forward citations	Standard essential patent (SEP)	Forward citations	Forward citations, generality, originality	Trading quality	Firm market value
Sample Size[i]	Train	150000	42736	4312	28286	28286	260	272
	Test	50000	10684	2154	7071	7071	101	70
	Years	2000-2005	1976-1991	-	2000-2009	2000-2009	-	1997-2006
	Area	General	Telecomms	IoT	Drugs	Drugs	Semiconductors	Pharmaceutical
Data Type	Numeric	x	x	x	x	x	x	x
	Categoric	x	-	x	-	-	-	-
	Text	-	-	-	-	Keywords	-	-
	Features	24	9	11	18	611	-	5
Parameter optimisation[j]	Hidden Neurons	512	200	10	4	-	-	3
	Layers	4	4	3	2	-	-	2
	Epochs	10000	Early stopping	6000	1000	-	10000	100000
Evaluation metrics[k,l]	Classes	2	-	2	4	4	2	-
	MAE	-	0.98	-	-	-	0.29	0.027
	RMSE	-	-	-	-	-	-	0.051
	Accuracy	# 0.71	-	0.73	0.91, 0.84, 0.71	# 0.61	0.85	-
	Precision	0.65	-	-	0.77, 0.51, 0.45	# 0.53, 0.55, 0.40	-	-
	Recall	0.92	-	-	0.37, 0.34, 0.37	# 0.29, 0.32, 0.30	-	-
	F1-score	# 0.76	-	-	# 0.50, 0.41, 0.41	# 0.37, 0.40, 0.34	-	-
	F2-score[m]	0.85	-	-	-	-	-	-

[a]The table is arranged in descending order, i.e. the most recent article is found on the left.

[b]The table is a subset of Table 2.11, and is synthesized from Tables 2.7 and 2.10.

[c]Empty cells correspond to non-existing values in the literature. The list of articles is not exhaustive.

[d]Learning paradigms represent the way and purpose of learning. There are 3 learning paradigms for AI: supervised, unsupervised, and reinforcement (Goodfellow et al., 2016; Murphy, 2012).

[e]Within the learning paradigm, the type indicates the type of model build for the application.

[f]Methodology refers to AI approaches deployed, with artificial neural networks (ANN) being the most popular approach.

[g]This is the algorithm used to evaluate the error-function derivative, with the backpropagation (BP) remaining the central learning algorithm (Schmidhuber, 2015).

[h]An output variable is a variable being predicted in supervised learning. In statistics, this is also known as the dependent variable, or target variable.

[i]Sample size refers to the split and number of datapoints between the training set and the testing set for the algorithms (see 4.5.1). We also include the technological area the dataset is limited and the year constrain.

[j]Parameter optimisation refers to the hyperparameters for neural networks. This is explained in Chapter 4 and for simplicity we report the most common hyperparameters.

[k]Evaluation metrics evaluate the model's generalising ability. Regression evaluation metrics include mean absolute error (MAE), mean absolute percentage error (MAPE), root mean square error (RMSE). Classification evaluation metrics include accuracy, precision, recall, and F1-score (see 4.3). For simplicity, not all evaluation metrics are reported.

[l]We have calculated all evaluation metrics in the table cells with the symbol #. These were calculated using information from the articles for comparative purposes.

[m]The Fbeta-score is a generalization of the F-score that adds a configuration parameter called beta. A default beta value is 1.0, which is the same as the F1-score. A beta value of 2, i.e. F2-score, gives more weight to recall and less weight to precision. Brownlee (2020a), URL: https://machinelearningmastery.com/fbeta-measure-for-machine-learning/

2.2.2.3.2.3 Parameter optimisation

From Table 2.12, while all the studies deploy the ANN methodology, all the studies have low capacity[1] models. About 75% are shallow neural networks[2], with only two recent studies delpoying a slightly deeper neural network, with 4 layers (Choi et al., 2020; Noh & Lee, 2020). In addition, 75% are narrow neural networks, i.e. have a narrow width[3], with only two studies deploying a slightly wider layer (Choi et al., 2020; Noh & Lee, 2020). All the models require a large number of epochs to be trained, despite their low capacity.

2.2.2.3.2.4 Evaluation metrics

The studies in Table 2.12 use a variety of evaluation metrics (see 4.3). Regression models use measures such as the mean absolute error (MAE) and root mean square error (RMSE) to evaluate the generalising ability of the model. With MAE of 0.98 and 0.29, the models proposed by Noh & Lee (2020) and Trappey et al. (2012) are partly driven by the low number of input features.

Classification models use measures such as the accuracy, precision, recall and F1-score. For these models, the accuracy ranges from 0.71 (Choi et al., 2020) - 0.91 (Lee et al., 2018). Precision, being the fraction of relevant instances among the retrieved instances, ranges from 0.40 (Woo et al., 2019) - 0.77 (Lee et al., 2018). Recall, being the fraction of the total amount of relevant instances that have been retrieved, ranges from 0.29 (Woo et al., 2019) - 0.92 (Choi et al., 2020). These models are able to identify lower value patents better than higher value patents. Choi et al. (2020) evaluate their models using a balanced dataset approach, where the 2 classes (maximum renewal vs. not renewal) are balanced, with the F2-score[4]. In addition, from the 5 articles proposing classification models, 3 are structured as 2-class tasks and 2 are structured as a 4-class task. Choi et al. (2020), a 2-class task, outperforms the other studies on recall, F1-score and F2-score; however, Lee et al. (2018), a 4-class task outperforms the other studies on accuracy and precision.

[1]The capacity of a neural network is defined as configuration of neurons or nodes and layers, i.e. the number of layers, the number of input nodes, the number of output nodes, and the number of nodes in each layer (Brownlee, 2019g; Hopfield, 1982; Jia et al., 2016).

[2]Shallow neural networks are artificial neural networks with 2 or 3 layers, i.e. 1 or 2 hidden layers. Deep neural networks are defined as networks with architectures with multiple hidden layers (Delalleau & Bengio, 2011; Goodfellow et al., 2016; Murphy, 2012).

[3]The number of input layer neurons and hidden layer neurons is referred to as the width, and the number of layers is referred to as the depth, of the neural network (Abood & Feltenberger, 2018; Brownlee, 2019g).

[4]The F2-score gives more weight to recall by arguing that a false negative rate is more critical than a false discovery rate in a practical business environment.

Chapter 3

Developing the Dataset

In this chapter, we focus on the process of developing the large dataset that is used in chapter 4 for developing the deep learning approach[1]. Fig. 3.1 shows the process flowchart for the dataset development, which is supported by computational resources[2] (see 1.2 and Fig. 1.1).

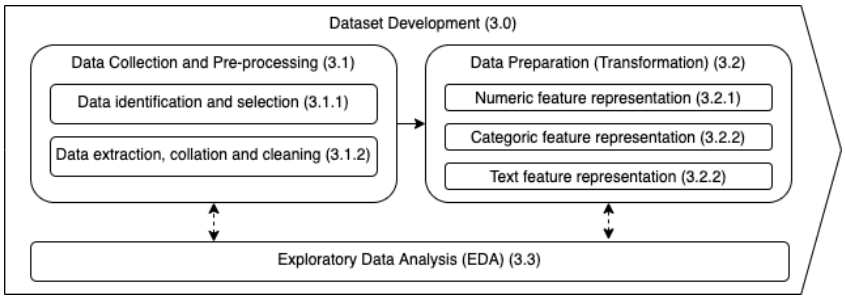


Fig. 3.1 Process flowchart of dataset development (a subset of Fig. 1.1)

Firstly, we describe and explain how the data are identified, selected, extracted, collated, and cleaned (3.1). Then, we describe the data preparation process (3.2), which involves the data transformation process of the numeric, categoric and text field values into features, and a feature representation dataframe. The feature transformation directly influences the model results because the success of all AI algorithms depends on how you present the data[3] (Liu & Motoda, 1999). We give a brief overview of the exploratory data analysis (EDA) continuously performed in 3.3.

[1]The code is written in Python language (Van Rossum & Drake, 1995), and uses the libraries of Tensorflow (Abadi et al., 2016a) and Keras (Chollet & Others, 2015). Tensorflow (Abadi et al., 2016b), URL: https://tensorflow.org. Keras (Chollet & Others, 2015), URL: https://keras.io.

[2]The data is stored in the cloud and processed with virtual machines using Microsoft Azure (Microsoft, 2020) and Google AI Platform servers (Google, 2020a). The code is written in Python (Van Rossum & Drake, 1995), and is stored and maintained on GitHub (Github, 2020).

[3]Koehrsen (2018c), URL: https://towardsdatascience.com/feature-engineering-what-powers-machine-learning-93ab191bcc2d

3.1 Data collection and pre-processing

We describe the process of data collection and pre-processing: (i) the data identification and selection (3.1.1), and (ii) data extraction, collation and cleaning (3.1.2). Fig. 3.2 gives an overview of the data collection and pre-processing process.

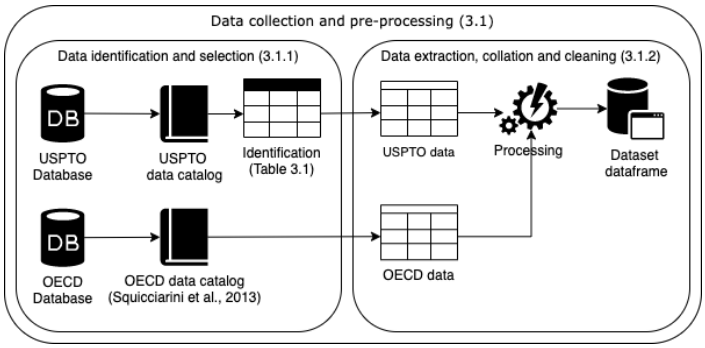


Fig. 3.2 Data collection and pre-processing process flow diagram (a subset of Fig. 3.1)

We use 2 main data collection sources: firstly, the USPTO[1] serves as our primary data collection source, through PatentsView[2]. The US is the world's largest patent market, where the majority of USPTO patents are also submitted in other countries (Bass & Kurgan, 2010; Lee et al., 2013; WIPO, 2019a,c, 2020). The USPTO database is well-organised, with high data quality and holds historical information (Cai & Zhu, 2015; Lee et al., 2017). Secondly, we use the OECD public data source[3], to complement the USPTO data.

3.1.1 Data identification and selection

We build a large dataset from 2 main data collection sources. Our primary data are sourced from the USPTO PatentsView database[4], and are complemented by the OECD patent quality indicators database[5]. From the USPTO data catalog (see Table 3.1), we select the following tables for extraction: application, brf_sum_text, claim, cpc_current, ipcr, maint_fees_events[6],

[1]United States Patent and Trademark Office (USPTO), URL: https://www.uspto.gov/.

[2]PatentsView, URL: https://www.patentsview.org/web/#viz/locations.

[3]Organisation for Economic Cooperation and Development (OECD), URL: https://data.oecd.org/.

[4]PatentsView, https://www.patentsview.org/download/.

[5]Organisation for Economic Cooperation and Development (OECD), URL: ftp://prese:Patents@ftp.oecd.org/Indicators_202001/. The OECD patent quality indicators database proposes a number of indicators to capture the quality of patents, intended as the technological and economic value of patents, and the possible impact that these might have on subsequent technological developments. It has been compiled for the OECD report by Squicciarini et al. (2013), and it contains respectively 16 patent value proxies and determinants at the USPTO patent level.

[6]United States Patent and Trademark Office (USPTO), URL: https://developer.uspto.gov/product/patent-maintenance-fee-events-and-description-files.

patent, uspatentcitation, together with the description, data element names, definition, example, and type of available data, for all tables. From the OECD data catalog (Squicciarini et al., 2013), we source all the data. In addition, for the USPTO data we restrict the dataset between the start date 01.01.1976 and end date 01.01.2019, and to patent kind codes 'A', 'B1', and 'B2', which represent granted patents[1]. We then drop all empty fields from the original USPTO source data of Table 3.1.

3.1.2 Data extraction, collation and cleaning

Following the data identification and selection in 3.1.1, we extract the data in CSV format in the cloud. Using a python developed automated script[2], we collate the data into a Pandas[3] dataframe, which is then saved as a dataset pickle[4] file (McKinney, 2010)[5,6,7]. As part of the collation process, data cleaning occurs where symbols and duplicates are removed. Data fields that are supposed to be empty are removed, where as data fields that are supposed to contain data are populated with zeros if empty. We save the dataset dataframe in the cloud.

[1]United States Patent and Trademark Office (USPTO), URL: https://www.uspto.gov/learning-and-resources/support-centers/electronic-business-center/kind-codes-included-uspto-patent. The USPTO began printing the WIPO Standard ST.16 code (kind code) on each of its published patent documents, which is used to distinguish the kind of patent document. Kind code 'A' represents granted patents up to 2001, which was later replaced by kind codes 'B1', 'B2'.

[2]The automated python script is a script that automatically compiles the dataset based on the constrains, i.e. the start date, end date, kind codes, and the USPTO data tables, such as the ones in Table 3.1. The script code then restructures the data: firstly, each table is deconstructed in columns and attached to a temporary dataframe, according to the patent id, which is used as a key. Secondly, the data is then cleaned by removing empty fields, and fields with wrongly introduced symbols or data types. Then, we calculate and operationalise all the variables, as shown in Table 4.1. Finally, the script restructures and re-orders the updated columns, to compose a permanent dataframe, which is saved in the cloud (Aristodemou, 2020a,b).

[3]The Pandas Development Team (2020), URL:https://pandas.pydata.org/. Pandas is a software library written for Python (Van Rossum & Drake, 1995) for data manipulation and analysis, offering data structures and operations for manipulating tables and time series.

[4]Van Rossum & Drake (2009), URL: https://docs.python.org/3/library/pickle.html. Pickle implements binary protocols for serializing and de-serializing a python object structure (Pandas dataframe), and its hierarchy is converted into a byte stream.

[5]In parallel, as a confirmatory process, we also use Google BigQuery with the same constrains as the Python script, to extract the USPTO data from the Google Patents Public dataset. The Google BigQuery platform allows for fast SQL queries, with standardised means of access, and we use it to confirm the total number of granted patents extracted.

[6]BigQuery on Google Cloud Platform, URL: https://cloud.google.com/bigquery. BigQuery is a fully-managed data warehouse that enables scalable, and fast analysis of big data working in conjunction with Google Cloud Storage.

[7]Google (2020a) Google Patents Public Datasets, URL: https://cloud.google.com/blog/products/gcp/google-patents-public-datasets-connecting-public-paid-and-private-patent-data.

Table 3.1 USPTO data catalog table identification

Table[a,b]	Description	Data Element[c]	Definition	Example	Type[d]
Application	Information	id	Application id assigned by USPTO	02/002761	varchar(36)
	on the	patent_id	Patent number	D345393	varchar(20)
	applications	series_code	Application series	2	varchar(20)
	for granted	number	Unique application	2002761	varchar(64)
	patent.	country	Country this application was filed in	US	varchar(20)
		date	Date of application filing	21/12/1992	date
Brf_sum_	Summary	patent_id	Patent number	8918554	varchar(20)
text	patent text	text	Text of the summary itself	Background	text
Claim	Full text of	patent_id	Patent number	4968079	varchar(20)
	patent	text	Claim text	A golf ball retriever...	text
	claims	dependent	Sequence number of dependent claims	1	int(11)
		sequence	Order of patent claims	10	int(11)
Cpc_	CPC classi-	patent_id	Patent number	3930271	varchar(20)
current	fication of	section_id	Cpc section	A	varchar(10)
	the patent[e]	subsection_id	Cpc subsection	A63	varchar(20)
		group_id	Cpc group	A63B	varchar(20)
		subgroup_id	Cpc subgroup	A63B71/146	varchar(20)
		category	Cpc category (primary or additional)	inventional	varchar(36)
		sequence	Order in which cpc class appears	0	int(11)
Ipcr	International	patent_id	Patent number	D409748	varchar(20)
	Patent Clas-	section	Ipc section	H	varchar(20)
	sification[f]	ipc_class	Ipc class	21	varchar(20)
		subclass	Ipc subclass	L	varchar(20)
		main_group	Ipc group	21	varchar(20)
		sequence	Order in which ipc class appears	0	int(11)
Patent	Data	id	Patent Id	3930271	varchar(20)
	concerning	type	Category of patent	utility	varchar(100)
	granted	number	Patent number	3930271	varchar(64)
	patents	country	Country in which patent was granted	US	varchar(20)
		date	Date when patent was granted	06/01/1976	date
		abstract	Abstract text of patent	A golf glove is...	text
		title	Title of patent	Golf glove	text
		kind[g]	ST.16 code	A	varchar(10)
		num_claims	Number of claims	4	int(11)
uspatent	Citations	patent_id	Patent number	9009250	varchar(20)
citation	made to US	citation_id	Patent to which select patent cites	8127342	varchar(20)
	granted	date	Date select patent (patent_id) cites patent (citation_id)	01/02/2012	date
	patents by	name	Name of cited record	Boynton et al.	varchar(64)
	US patents	category	who cited the patent (examiner, applicant)	cited by patent	varchar(20)
		sequence	order of the citation	622	int(11)
WIPO	-	field_title	WIPO technology field title	Electrical machinery	varhcar(255)

[a]The USPTO data catalog table identification, where the data elements are defined with examples and the type of data. The list is not exhaustive, and only a selected number of data elements are reported.

[b]The table names are kept the same as found in the USPTO data catalog.

[c]The data element represents the particular patent field, feature or cell data available.

[d]The type shows the type of the data structure and in brackets its length. For example, 'varchar' represents variable character, and 'int' represents integer.

[e]United States Patent and Trademark Office (USPTO), URL: https://www.uspto.gov/web/patents/classification/cpc/html/cpc.html.

[f]World Intellectual Property Organisation (WIPO), URL: https://www.wipo.int/classifications/ipc/en/.

[g]United States Patent and Trademark Office (USPTO), URL: https://www.uspto.gov/learning-and-resources/support-centers/electronic-business-center/kind-codes-included-uspto-patent.

3.2 Data preparation (transformation)

In this section, we describe the process of the data preparation (transformation). The transformation involves the transformation of the numeric, categoric and text data, into features[1] to form a features dataframe. This is also known as feature engineering[2], which is the process of transforming data into features that represent the underlying problem to the predictive models (Domingos, 2012; Liu & Motoda, 1999). The success of all AI algorithms depends on how you present the data to the predictive model, i.e. the transformed representation[3].

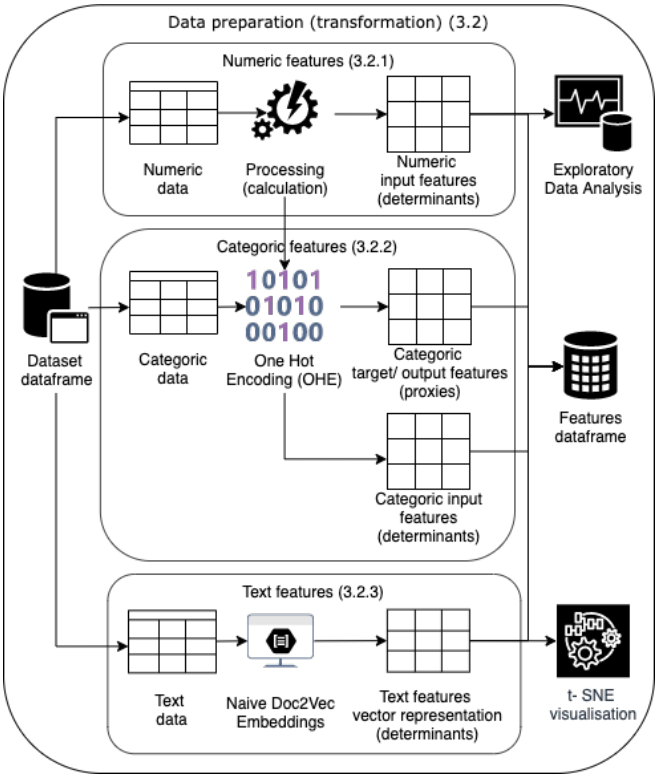


Fig. 3.3 Data preparation (transformation)

Fig. 3.3 shows an overview of the data preparation (transformation) process. Patent data

[1] In artificial intelligence (AI), a feature is a variable, an individual characteristic of a phenomenon being observed, which can be quantified (Bishop, 2006). For example, for this research, the number of backward citations is a numeric input feature, i.e. a numeric feature patent value determinant, where as forward citations are a categoric target or output feature, i.e. a categoric feature patent value proxy. Thus, input features are patent value determinants, and target/ output features, are patent value proxies (see Table 4.1).

[2] Brownlee (2014), URL: https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/.

[3] Koehrsen (2018c), URL: https://towardsdatascience.com/feature-engineering-what-powers-machine-learning-93ab191bcc2d.

are a series of structured and unstructured data[1]. Structured patent data include numeric data, i.e. the citation information, and categoric data, i.e. the IPC/ CPC classifications. Unstructured patent data includes the narrative text (abstract, claims, summary, title) (Abbas et al., 2014).

Firstly, we transform the numeric data, into numeric features (3.2.1). Using the one-hot-encoding (OHE) methodology, we perform 2 transformations: (i) the categoric data, i.e. IPC/ CPC classifications, into categoric features (3.2.2) and particularly the categoric input feature determinants (3.2.2.2); (ii) the continuous numeric features, i.e the grant lag index, the forward citations, the generality index, and the patent renewals, into categoric target feature proxies (3.2.2.3)[2]. The text data is transformed into text features (3.2.3) with our developed Doc2Vec methodology and visualised using the t-SNE[3] algorithm (Van Der Maaten & Hinton, 2008).

3.2.1 Numeric feature representation

Loading the dataset into a Pandas dataframe, we process all numeric data, transforming them into numeric features (see Fig. 3.3). Firstly, we process the numeric data into the numeric input features, by calculating the patent value determinants (2.1.2.3). These include: the number of claims, number of independent claims, the number of independent claims, the patent scope, the number of backward citations , the number of Non-Patent Literature references, the originality index, the radicalness index, the family size, the technology field, the relevance of classification to many fields, the number of primary unique CPC invention sections, the number of primary unique CPC invention subsections, the number of primary unique CPC invention main groups, the number of primary unique CPC invention subgroups, the number of additional unique CPC invention sections, the number of additional unique CPC invention subsections, the number of primary unique CPC additional main groups, the number of additional unique CPC invention subgroups (see Table 4.1).

Secondly, we process the numeric data into the numeric target/ output features, by calculating the patent value proxies (2.1.2.3). These include: the grant lag index, the forward citations, the generality index, and the patent renewals. We align the calculation of the foward citations and the generality index, with the patent renewal timeline, i.e. in T years

[1]Structured data is data with clearly defined data types whose pattern makes them easily searchable. Unstructured data is data with no underlying structure or pattern (Lupu, 2013; Lupu et al., 2011; Manning et al., 2008).

[2]These transformations result in structuring the problem in a supervised classification approach (see 4.1).

[3]The t-SNE algorithm is a non-linear dimensionality reduction approach, suited for embedding high-dimensional data for visualization in a low-dimensional space of 2 or 3 dimensions. It models similar objects by nearby points and dissimilar objects by distant points. We use the t-SNE algorithm to visualise the patent text sections and visually evaluate the effectiveness of our developed Doc2Vec methodology (3.2.3.3.2).

after the grant date, where $T = \{4, 8, 12\}$. We also include the generality and patent quality index 4, as calculated by the OECD (Squicciarini et al., 2013). We then transform these continuous numeric target/ output features, into categoric target/ output feature proxies, using one-hot-encoding (OHE) (see 3.2.2).

3.2.2 Categoric feature representation

3.2.2.1 One-hot-encoding methodology

Artificial intelligence (AI) algorithms cannot work with categoric data directly. Categoric data are variables that contain label values rather than numeric values. Categoric data must be converted to numbers in order to be able to work on AI algorithms[1]. This involves a 2-step process: firstly the categorical data are assigned an integer value using a label encoder[2], and secondly an one-hot encoding (OHE) is applied to the integer representation, where the integer encoded variable is removed and a new binary variable is added for each unique integer value[3] (Murphy, 2012). Fig. 3.4 shows an example of one-hot-encoding (OHE) of the IPC section variables. There are 8 IPC sections, identified from A to H. The label set S , of size $[8x1]$, consists of 8 labels $S = [A, B, C, D, E, F, G, H]$. The label set is then converted to a numeric set, S_{le}, of size $[8x1]$, using a label encoder $S_{le} = [0, 1, 2, 3, 4, 5, 6, 7]$, which in turn is transformed to an orthogonal matrix of size $nx[1x8]$, where n is the examples.

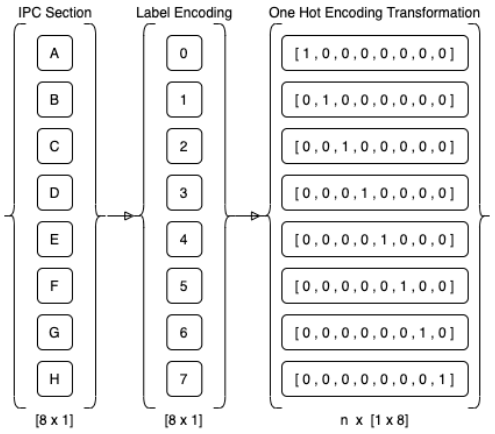


Fig. 3.4 One hot encoding (OHE) transformation example

[1] Brownlee (2017h), URL: https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/.

[2] The integer values have a natural ordered relationship between each other and algorithms may be able to understand and harness this relationship. The problem is that with label encoding, the categories now have natural ordered relationships. The computer is programmed to treat higher numbers as higher numbers, and thus it will naturally give the higher numbers higher weights, which may result in poor performance .

[3] Brownlee (2017f), URL: https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/.

3.2.2.2 Categoric input features (determinants) representation

Firstly, we process the categoric data, focusing on patent value determinants (2.1.2.4). These include the IPC and CPC classifications[1], to transform them into categoric input feature determinants. We one-hot-encode the CPC section, CPC subsection, CPC main group, IPC section and IPC class. Due to the language proximity (3.2.3.3.2), the IPC subclass and IPC main group of the classification are covered by the CPC main group[2].

3.2.2.3 Categoric target/ output features (proxies) representation

We then focus on transofrming our target/ output features. Following 3.2.1 and Table 2.4, we transform the continuous numeric/ target output features, into categoric target/ output feature proxies. From 2.2.2.3.2, we identify that the majority of articles are structured as supervised classification approaches. Thus, we structure our problem, of identifying valuable patents, into a supervised classification approach (see 4.1). The categoric target/ output feature proxies represent patent value on one or more value dimensions (Table 2.3). These are structured as a binary classification, where class V_H represents a high value patented invention, and a V_L represents a low value patented invention. We use a binary classification[3], for the following reasons: (i) the model interpretability[4], (ii) simplicity and practicality, (iii) the model complexity[5], and (iv) previous studies on patent value with AI methodologies and patent data have used, in the majority, a binary classification with higher evaluation performance (see 2.2.2.3.2.4 and Table 2.12).

Table 3.2 shows the categoric output features proxies, with the output variables (deployed in chapter 4), the definition of the two value classes, and the justification. These include:

[1]For the transformation, the categoric data includes the IPC and CPC classifications with the sections, classes, subclasses, main groups and subgroups.

[2]We include the higher levels of the IPC classification, i.e. the section and class, to cover the international presence of the patent, but we use the lower levels of the CPC classification, i.e. the main group, to capture the class presence of the patent. This is because the CPC classification is a joint endeavour of the European Patent Office (EPO) and the United States Patent and Trademark Office (USPTO) to harmonize their classification systems into a single system having a similar structure to the International Patent Classification (IPC) administered by WIPO (EPO, 2017). The jointly developed classification system is more detailed than the IPC to improve patent searching. The IPC classification is not available for USPTO data before 2005.

[3]This is similar to one-vs-all classification scheme. Brownlee (2020d), URL: https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/.

[4]Pagels (2018), URL: https://medium.com/value-stream-design/machine-learning-reductions-mother-algorithms-part-ii-multiclass-to-binary-classification-1dad599147b.

[5]A binary class represents one pair of classes and it's possible to understand more about the association factors (Har-Peled et al., 2002). The lower the number of output nodes (i.e. classes) the lower the model complexity, which makes the likelihood inference of the model higher for those output nodes. Given a fixed amount of data, a greater number of output nodes will lead to poorer results (Lorena et al., 2008; Tewari & Bartlett, 2007).

the grant lag index, the forward citations, the generality index, and the patent renewals. We align the calculation of the forward citations and the generality index, with the patent renewal timeline, i.e. in T years after the grant date, where $T = \{4, 8, 12\}$. We also include the generality and patent quality index 4, as calculated by the OECD (Squicciarini et al., 2013).

Table 3.2 Categoric output/ target feature proxies class definition

Categoric output[a]	Output variable[b]	Class[c,d]		Definition	Percentile[e]	Justification
		V_H	V_L			
Forward Citations	Citations_t4, Citations_t8, Citations_t12	21> citations	<20 citations	Hall (2005)	>75th	Firms with patents with forward citations of more than 20 citations per patent, have a 54% higher market value of what would be expected, given their R&D capital and patent stock
Grant lag	Grant_Lag	<600 days	600> days	Harhoff & Wagner (2009)	>75th	The shorter the time between application and granting, the higher the value of the patent. Patents with time less than 600 days, are considered valuable
Generality Index	Generality_t4, Generality_t8, Generality_t12, Generality	0.75>	<0.75	Aristodemou & Tietze (2018b)	>75th	Patents with a generality of higher than 0.75 have a diversified number of forward citations influencing a diverse range of technology fields
Renewals	Renewal_t4, Renewal_t8, Renewal_t12	Renewed	Not Renewed	Lanjouw et al. (1998)	>75th	Firms renew their most valuable patents, and also keep alive the ones that are strategically and economically important to them
Patent Quality Index 4	Quality_Index_4	0.45>	<0.45	Squicciarini et al. (2013)	>75th	The index considers the following compoentns: number of forward citations (up to 5 years after publication), patent family size, number of claims, and the patent generality index, with equal weights. It is can be considered exceptional above 0.45, but should be interpreted with care (Lanjouw & Schankerman, 2004)

[a]The patent value proxies with their definitions, rational and patent value dimension can be found in 2.1.2.2, and Table 2.4. The table is arranged in alphabetical order according to the proxy.

[b]Output variables are the variables being predicted by the model (see Table 4.1).

[c]Binary classification patent value classes, where class V_H represents a high value patented invention, and a V_L represents a low value patented invention.

[d]The dataset consists of the full USPTO granted patents from 1976-2019 (see 3.1.2), and the model development (see chapter 4) considers all the population. In principle, focusing only on granted patents and not including patent applications can introduce survival bias (Brown et al., 1992; Deng et al., 1999; Groeneveld & Meeden, 1984; Lin & Chen, 2005; Seru, 2014); however, granted patents provide exclusive market rights to the firms for exploitation, which gives an indication of the market value of inventions (Hall, 2005).

[e]Percentile of high value distribution of the feature, i.e. a patent with 21 forward citations belongs to the 75th percentile and above in the value distribution.

For the patent value proxies in Table 2.4, we define a cut-off threshold, the separation between high value, V_H and low value, V_L, based on previous literature. Scholars have argued about these cut-off thresholds on what constitutes high value and a low value patent and these are justified in Table 3.2. For forward citations, the cut-off threshold is 21 citations, where Hall (2005) argues that firms with patents with forward citation of more than 20 citations per patent, have a 54% higher market value. A patent with more than 21 citations is classed as high value V_H, and otherwise as low value V_L[6].

[6]From Table 3.5, it is evident that the cut-off threshold of 21 citations is 7 times more than the 75th percentile for citation_t4, 3 times more than the 75th percentile for citation_t8, and about 2 times the 75th percentile for citation_t12. All distributions are positively skewed, with the Pearson's first skewness coefficient (mode skewness) of 0.24 for citation_t4, 0.26 for citation_t8 and 0.20 for citation_t12. In addition, the Yule's

For renewals, the cut-off threshold is if a patent has been renewed, because firms renew their most valuable patents and keep alive the ones that are strategically and economically important to them (Lanjouw et al., 1998)[1]. For grant lag, the cut-off threshold is 600 days, with a shorter time between application and granting indicating a high value patent (Harhoff & Wagner, 2009). For generality, the cut-off threshold is 0.75, giving a diverse range of technology fields in the forward citations (Aristodemou & Tietze, 2018b)[2]. The patent quality index 4, as defined by Squicciarini et al. (2013), has a cut-off threshold of 0.45, with high value patents V_H considered exceptional if it's higher[3]. Thus, for every output variable, we form 2 classes, the high value V_H and low value V_L patents, based on cut-off thresholds of what constitutes a high/low value patent in the literature[4].

3.2.3 Text feature representation - Naive Doc2Vec

Patent text has long be considered a patent value determinant (Lanjouw & Schankerman, 2004) and has been associated with the technological and economical value of patents (Table 2.5). Patents contain four main sections of text: abstract, claims, summary/ description, title. Several scholars have used the patent text for patent information retrieval (Table 2.9), mainly

coefficient of skewness is 0.33 for citation_t4, 0.33 for citation_t8 and 0.40 for citation_t12. The positive skewed distribution follows the distribution identified by Hall (2005). Older patents in t8 and t12 accumulate forward citations with age. Given that the number of datapoints is restricted to the patent reaching age 8 and 12, survival bias can be introduced given that the patents that have not reached the required age are excluded (Brown et al., 1992; Groeneveld & Meeden, 1984). However, all patents remaining are treated similarly, with the model taking into consideration high value, V_H and low value, V_L patents at the same point in time. This was introduced to ensure that there is a complete set of input feature determinants and output proxies for every patents (see 4.5).

[1]From Table 3.5, the renewals (t4, t8, t12) distribution is negatively skewed, with the Pearson's first skewness coefficient (mode skewness) of -0.15, and the Yule's coefficient of skewness of -0.11. This is similar to the distribution by Squicciarini et al. (2013), with the majority of patents being renewed by firms in the fear of losing out.

[2]From Table 3.5, it is evident that the cut-off threshold of 0.75 is 2.9 times more than the 75th percentile for generality_t4, 1.5 times more than the 75th percentile for generality_t8, and about 1.3 times the 75th percentile for generality_t12. All distributions are positively skewed, with the Pearson's first skewness coefficient (mode skewness) of 0.56 for generality_t4, 0.83 for generality_t8 and 0.30 for generality_t12. In addition, the Yule's coefficient of skewness is 1.00 for generality_t4, 1.00 for generality_t8 and 0.33 for generality_t12. The relative increase in forward citations is less than the diversification in the IPC classes, showing that there is a small number of patents covering a diverse number of fields, with increasing diversification as the number of forward citations increases (i.e. forwarded cited patents are filed in a diverse number of fields).

[3]From Table 3.5, it is evident that the cut-off threshold of 0.45 is 1.3 times more than the 75th percentile. The distributions is close to an asymmetric distribution, with the Pearson's first skewness coefficient (mode skewness) of 0.05, and the Yule's coefficient of skewness of -0.02. This is partly the reason for the improved model performance in Table 5.15.

[4]From a practical point of view, technology managers, intellectual property managers, and firms are interested on the valuable patented inventions (Choi et al., 2020), and if a patent is high value or low value (Altuntas et al., 2015; Poege et al., 2019).

focusing on classifying patents into respective technological classes, based on the frequency of words (Abbas et al., 2014). Recently, some scholars have used more advanced methods of natural language processing (NLP) to capture the semantic and contextual meanings of words for technological area classification. However, a limited number of studies have used the patent text as input to artificial intelligence (AI) methods for valuation purposes (Table 2.12), with many studies focusing on numeric and categoric types of data (see 2.2.2.3.2.2).

Firstly, we review the literature using the narrative approach (Cronin et al., 2008; Paré et al., 2015) on natural language processing (NLP) vector space models (VSM) that capture semantic and contextual similarity (3.2.3.1). We then provide an overview of how these models have been used with patent data (3.2.3.2). Secondly, we describe the text feature vector representation method[1] (3.2.3.3), which consists of two parts: (i) the development of the Doc2Vec methodology, for representing patent text into vector input features (3.2.3.3.1); (ii) the visual representation with t-SNE approach of the Doc2Vec vectors (3.2.3.3.2).

3.2.3.1 Natural language processing (NLP) vector space models (VSM)

Text data can be vectorised or transformed into a vector, which captures the complex relationships that exist within the text[2] (syntactic, semantic, sequential) (Jurafsky & Martin, 2016; Lupu, 2013). There are two overlapping groups of methods deployed for the analysis of text (Abbas et al., 2014; Jurafsky & Martin, 2016). The first group includes traditional document-term matrix representations such as the term frequency (TF), inverse document frequency (IDF), bag of words (BOW), dictionary-based approaches, and rule-based approaches. For example, some studies identified in Table 2.9 make use of specific chemical compound dictionaries to identify chemical elements in patents (Krallinger et al., 2015). The second group includes more advanced methods that are related to distributed text representation, which can be captured by vector space models (VSMs). Such representations or transformations can be captured with dense[3] vectors, where the vectors are short (relative to sparse vector representations) and dense (non-zero values) (Mikolov et al., 2013a). Some of these approaches consist of linear discriminant analysis (LDA), multi-dimensional scaling (MDS), principal component analysis (PCA), quadratic discriminant analysis (QDA), singular value decomposition (SVD). Scholars have used these approaches with patent data (Table 2.9), for

[1]This transformation is necessary to utilise the patent text as an input to the developed AI deep learning methodology of valuing patents in chapter 4.

[2]The success of all artificial intelligence algorithms depends on how you present the data to the predictive model, i.e. the transformed representation (Koehrsen, 2018c).

[3]Generally, dense vectors generalize better because they contain less parameters than sparse vectors, and can capture context and semantic relationships better, i.e. similar words are mapped to nearby points (Jurafsky & Martin, 2016).

example Jun et al. (2014) analyse technological trends by building a document-term matrix, combined with a K-means clustering method and dimensionality reduction.

Recent advancements in NLP compose distributed document text representations from word representations in vector spaces. Word embeddings is a method by which words are transformed to vectors using their semantic meaning. Mikolov et al. (2013a) develop the Word2Vec model, where the words and the context are captured in a vector transformation. This allows words with similar meanings to be clustered together and share relationships (Mikolov et al., 2013b). Such methods include the Word2Vec models, which consist of (i) the continuous bag of words (WV-CBOW) model, and (ii) the continuous skip-gram model (WV-CSG) (Mikolov et al., 2013a). Word2Vec models aim to predict a word based on the context regardless of its position (WV-CBOW) or predict the words that surround a given single word (WV-CSG) (Mikolov et al., 2013a). Word embeddings can be applied to larger corpora of text by taking the centroids of the individual vectors to generate a document embedding (Le et al., 2014), which can then be used in neural networks. This captures the sequential, contextual and semantic similarity of words, sentences and paragraphs (Dai et al., 2015). Document embedding methods include the paragraph vector models, which consist of (i) the distributed memory paragraph vector (PV-DM), and (ii) the distributed bag of words of paragraph vector (PV-DBOW).

Fig. 3.5 shows the representations of the Word2Vec and paragraph vector models[1] model, as defined by Mikolov et al. (2013a) and Le et al. (2014). The WV-CBOW, given a window of words, from $w(t-2)$ to $w(t+2)$ can predict the next word $w(t)$ in the sequence from the syntactic and contextual similarity. The WV-CSG, given a word $w(t)$, with a syntactic and contextual representation, can predict the neighbouring words, with similar semantic meanings (Stein et al., 2019). The PV-DM, given a set of words W, keeps in the memory a paragraph D, which captures the syntactic and semantic similarity of the sentences, paragraphs or the document, and can predict the next. This is based on the similarity of the semantic representation, and paragraph vector D, which acts as a global representation. The PV-DBOW can predict a set of sentences, paragraphs or documents, given a paragraph vector D, which captures the semantic and syntactic representation[2].

[1]Eclipse Deeplearning4j Development Team (2020), URL: https://deeplearning4j.konduit.ai/language-processing/word2vec.

[2]The semantic and syntactic evaluation of Word2Vec models and vector space models (VSMs) involves the evaluation against a test developed by Google (Google Code, 2013; Mikolov et al., 2013a). This test measures how well the model captures the semantic and syntactic relationships between words. The model is presented with an analogy, which exists in the vocabulary of the model, such as 'King':'Queen'::'Man':'?'. This is solved by finding a vector x such that $vec(x)$ is closest to $vec('Queen') - vec('King') + vec('Man')$, according to the cosine distance. This example is considered correct if the model predicts x as 'Woman', which would represent the model's ability to infer the relationship. The task has two broad categories: the syntactic analogies (such as 'quick':'quickly'::'slow':'slowly') and the semantic analogies, such as the country to capital city relationship

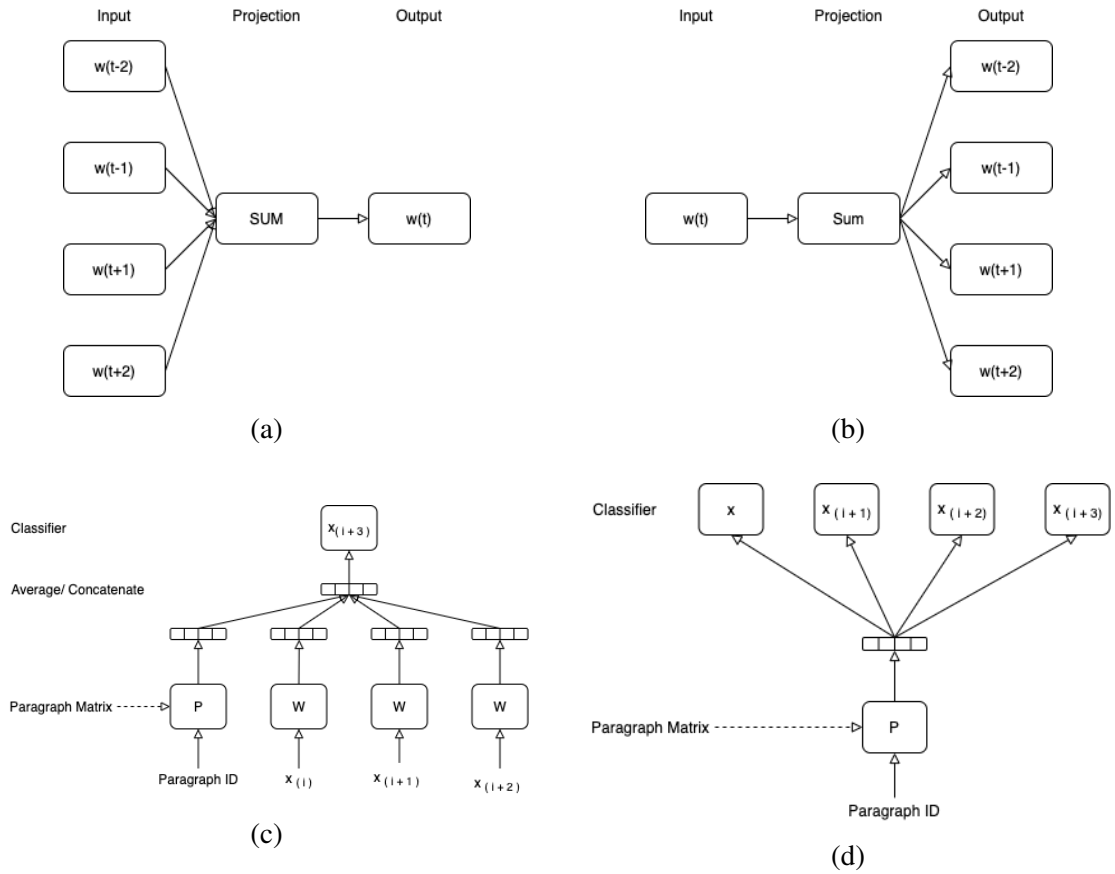


Fig. 3.5 Model representations of: 3.5a Word2Vec - continuous bag of words (WV-CBOW); 3.5b Word2Vec - continuous skip-gram (WV-CSG); 3.5c Paragraph vector - distributed memory paragraph vector (PV-DM); 3.5d Paragraph vector - distributed bag of words of paragraph Vector (PV-DBOW)

3.2.3.2 Vector space models (VSM) with patent text

The use of embeddings has recently gained considerable attention, with an increasing interest in using advanced NLP models to represent patent text data (Skripnikova, 2019). From 3.2.3.1 and Table 2.9, we identify relevant articles that deploy the vector space models (VSM) with patent data, using a narrative approach (Cronin et al., 2008). The majority of these articles are in the area of information retrieval, and limited studies have used the patent text for patent valuation (see Table 2.12). We review these articles to identify VSMs applications and limitations, to help us with the development of our VSM, Doc2Vec (3.2.3.3).

Table 3.3 shows the patent text VSM studies, with the majority published after 2018. All of the studies make use of pre-processing data approaches (see Table 2.9). All the articles identify similarity in patents and predict IPC/ CPC classifications, mainly used in

(Leonard, 2013).

prior art search. The studies are technology area specific with highest precision of 0.90 (Trappey et al., 2020b) and highest F1-score of 0.64 (Hu et al., 2018a). Hu et al. (2018a) propose a hierarchical feature extraction model for mechanical patents, which is able to capture local and global features of phrases including temporal semantics. Similarly, Lu et al. (2020) propose a similarity model evaluating text representation approaches (Lei et al., 2019). However, it is difficult to directly compare them because of different evaluation metrics. Govindarajan et al. (2019a) develop a topic model for topic generation (Trappey et al., 2012, 2020b), similar to Tenorio-González & Morales (2018), who develop a text automatic concept generation method (Hurtado et al., 2016; Jeong et al., 2018). Helmers et al. (2019) use a comparative method to evaluate different embeddings on patent prior art search. Abdelgawad et al. (2020) develop an algorithm for automatic classification of patent documents into IPC classes (Aras et al., 2018; Lee & Hsiang, 2019a,b). Hofstätter et al. (2019) explore the use of word embeddings for patent retrieval.

Table 3.3 Vector space model (VSM) studies with patent text

Author[a]	Algorithmic approach[b]	Output variable	Sample size[c]	Dataset area	Evaluation measure	Evaluation result
Lu et al. (2020)	Recurrent neural network (RNN)	Similarity	55615	Text processing, telecommunications	Mean absolute error (MAE)	0.09
Trappey et al. (2020b)	Sequence to sequence with attention (SSWA)	Patent summary	1708	AI techniques in smart machinery	ROUGE	Precision: 0.90 Recall: 0.84
Govindarajan et al. (2019a)	Excessive topic generation (ETG), Latent semantic analysis (LSA)	IPC/ CPC classification	741	IoT	Accuracy	0.8
Helmers et al. (2019)	Bag of words (BOW), Latent semantic analysis (LSA), kernel principal component analysis (KPCA), Word2Vec continuous bag of words (WV-CBOW), Paragraph vector (Doc2Vec)	Similarity	2500	Medical	Area under curve (AUC)	-
Lei et al. (2019)	Latent dirichlet allocation (LDA), Convolution neural network (CNN)	Similarity	8942	IoT	Euclidean distance	0.67-0.91
Hu et al. (2018a)	Hierarchical feature extraction model (HFEM), with BiLSTM and convolution neural network (CNN)	CPC classification	90000	Mechanical	F1 score, precision, recall	F1 score: 0.64 Precision: 0.82 Recall: 0.55
Jeong et al. (2018)	Topic modelling, Latent semantic analysis (LSA)	Topics, keywords	-	-	-	-
Trappey et al. (2012)	Kaiser–Meyer–Olkin (KMO) approach, Principal component analysis (PCA)	Trading quality	361	Semiconductors	Accuracy	0.85

[a]The studies are arranged in descending order, i.e. the most recent study is found at the top.
[b]The algorithmic approach is the methodology used to process the patent text.
[c]The sample size is the total number of patents for training and testing the models.

Although previous studies have used patent text data a number of them extracted features based on word frequency (Jeong et al., 2018; Lei et al., 2019; Trappey et al., 2012). Recently, several of the studies in Table 3.3 has focused on the semantic and contextual meaning of the text, or the sequential order of words. However, these have been in the information retrieval and prior art search. Limited studies have used the patent text or a patent text VSM for patent valuation (see Table 2.12). In addition, some studies have used patent text VSMs, such as Word2Vec or Doc2Vec, for comparative purposes (Helmers et al., 2019; Lu et al., 2020).

Consequently, it is necessary to derive features that have the patent documents' contextual meaning and capture the patents' technical content. These can be used in patent value models to improve prediction (see chapter 4).

3.2.3.3 Doc2Vec - vector space model (VSM) methodology

Embeddings are a learned representation of data, often using ANN architectures. They reduce the dimensionality of an input set by mapping the most important features of the set to a vector of continuous numbers. We develop and adapt a text vector space model (VSM) representation, based on the paragraph vector, Doc2Vec (Fig. 3.5). We do this to capture the syntactic and semantic relationships of the patent text sections, and convert them into a matrix, which can be processed by AI methodologies as input features (see chapter 4).

3.2.3.3.1 Doc2Vec vector feature representation

Doc2Vec[1] are distributed embedding vectors of sentences and paragraphs (Grzegorczyk, 2019), capable of constructing representations of variable length sequences (Dai et al., 2015; Le et al., 2014; Pagliardini et al., 2018). From Table 3.3, we have identified limited studies that use Doc2Vec with patent text data. Lu et al. (2020) run a comparative analysis of NLP methods to measure patent citation similarity, with different VSMs. One of these is the Doc2Vec, which significantly reduces the mean absolute error (MAE). However, their analysis seems to be dataset specific to the technological area of electronics, which partly drives some of their results. Similarly, Helmers et al. (2019) find that the Doc2Vec is more meaningful and effective on individual patent text sections, i.e. abstract, claims, because of the reduction in noise (Carvalho & Nguyen, 2017)[2].

To transform the patent text into a distributed embedding vector of sentences and paragraphs, which captures the syntactic and semantic relatioships[3,4], we propose an adapted Doc2Vec methodology (Le et al., 2014). Firstly, a vocabulary set v of size n is constructed by

[1]Doc2Vec has a few advantages over other methods, with the vector inheriting the semantic and syntactic properties of the words (Le et al., 2014). This ensures that the context of the document is maintained, maintaining the global document corpus context (influenced by N, the total number of documents) (Dai et al., 2015). The semantic relationship between the Doc2Vec dimension D exists because the proximity is maintained (Grzegorczyk, 2019; Le et al., 2014).

[2]Given the recency in all articles using the Doc2Vec methodology (Helmers et al., 2019; Lu et al., 2020), we follow some of the principles introduced. For example, for every patent section (abstract, claims, summary, title), we produce a separate Doc2Vec vector for each patent text secion, following the findings by Helmers et al. (2019). In our dataset, each patent document consists of 4 Doc2Vec vectors of 300 dimensions.

[3]Rehurek & Sojka (2010), Gensim Library ,URL: https://radimrehurek.com/gensim/index.html. Gensim is an open-source library for unsupervised topic modeling and natural language processing.

[4]Bird et al. (2009), Natural Language Toolkit (NTLK) Package, URL: https://www.nltk.org/. NLTK is a package, libraries for symbolic and statistical natural language processing for English.

extracting all words from the patent corpus (Eqn. 3.1a). Each word vector w contained in this vocabulary has a fixed dimension d, and is randomly initialized. Each patent document x consists of a subset of words belonging to v , and contains l_x words. The word representations are learned with the Word2Vec continuous bag of words (WV-CBOW) approach[1], which predicts the target word t_{ij} based on a window of context words c_{ij} of size k, i is the index number of a patent document, $j \in (1,l)$ is the index number of a word in patent document i (Eqn. 3.1b and 3.1c). We apply zero padding[2] up to a size of k after w_{ij}, to maintain the same window size for all words. To construct the embedding vectors for patent documents, we construct a randomly initialized embedding vector p_i. The vector u_i, which represents the patent document embedding vectors, is of dimension d and is initialized randomly. The embedding vectors are updated dynamically based on gradients (Eqn. 3.1d).

Fig. 3.6 shows the process we follow for our adapted Doc2Vec methodology, to transform the patent text sections into vector space model feature representations. Firstly, we make use of the pre-trained Word2Vec continous bag of words (WV-CBOW) model by Google[3]. The pre-trained model has been trained on 5.9 million US patent abstracts[4,5]. We use the extracted $[nxm]$ matrix of the model, where n is the number of words in the vocabulary and m is the size of the word embedding vector. The size of the vocabulary is comprised of 33 million specialised words. The size of the embedding vector is 300 dimensions[6].

$$v = \{w_1, w_2, w_3, ..., w_n\} \tag{3.1a}$$

$$t_{ij} = w_{ij} \tag{3.1b}$$

$$c_{ij} = \{w_{i(j-k)}, ..., w_{i(j+k)}, w_{ij} \notin c_{ij}\} \tag{3.1c}$$

$$p_i = \frac{1}{2}\left((u_i + \frac{1}{l}\sum_j c_{ij}\right) \tag{3.1d}$$

[1]Brownlee (2017d), URL: https://machinelearningmastery.com/develop-word-embeddings-python-gensim/.

[2]Zero padding refers to adding zeros to a vector to increase its length to specific size (Murphy, 2012).

[3]Google (2020b), URL: https://github.com/google/patents-public-data.

[4]Google Cloud (2019b), URL: https://console.cloud.google.com/storage/browser/patent_landscapes/.

[5]This is also comparable to our total dataset (see 3.3).

[6]The dimension determines the size of the resulting vector space, i.e. the vector size. Mikolov et al. (2013a) argue that increasing the model's dimensionality after a certain point leads to incremental improvements in model performance. Experimentation with other dimensions has shown degradation in model performance (Abood & Feltenberger, 2018), and thus we determine a dimension of 300 to be the most appropriate (Abbott, 2018; Habibi et al., 2017; Schakel & Wilson, 2015; Stein et al., 2019; Yin & Shen, 2018).

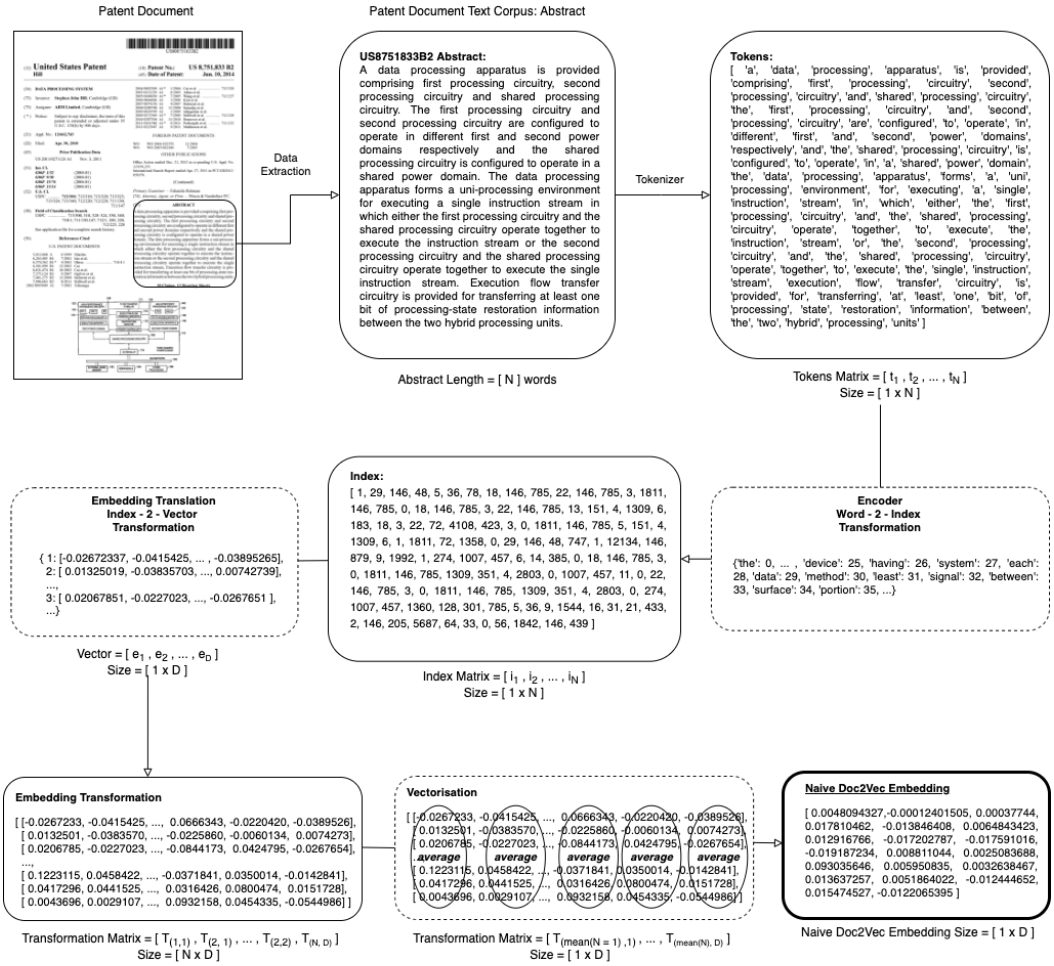


Fig. 3.6 Doc2Vec methodology for transforming the patent text into vector feature embedding

From previous studies (Helmers et al., 2019), we identify the patent abstract as a concise representative summary of the patented invention, which articulates the contents of the invention in the legal drafting language[1] (WIPO, 2004). Therefore, we use these pre-trained Google patent abstract word embeddings[2] as an approximation of the contextual representation of the semantic and syntactic relationships that exist within the patent text, and transform all the patent text sections[3].

From the data collation (3.1.2), we tokenise the patent text for every section (abstract, claims, summary, title) in every patent, which creates a tokens matrix of size N (the number

[1]The patent abstract is a concise representative summary of the patent, since it is important to act as a stand alone section, which can clearly articulate the contents of the invention in a legal drafting language. According to WIPO (2004), the principle is that the abstract should be as concise as the disclosure permits, since it serves as a scanning tool for searching purposes.

[2]Google Cloud (2019a), URL: https://cloud.google.com/blog/products/data-analytics/expanding-your-patent-set-with-ml-and-bigquery.

[3]Google (2019), URL: https://github.com/google/patents-public-data/tree/master/models/landscaping.

of words in a specific patent section per patent)[1]. We use an encoder, for the word-2-index transformation, which transforms the tokens matrix into an index matrix of size N. The encoder makes use of the pre-trained Google patent Word2Vec model's vocabulary n, where n is the number of words the vocabulary of 5.9 million US patent abstracts, to map the tokenised text into a sequence of numbers, which maintains the semantic and sequential positioning. Then, we use an embedding translation, for the index-2-vector transformation, which transforms the index matrix to an embedding transformation matrix of size $[NxD]$. This represents the total document vector, and using a vectorisation transformation, we concatenate/ average the contextual dimensions (representing the syntactic and semantic positions) to form the final Doc2Vec embedding vector representation of size $[1xD]$, for every patent section (see Fig. 3.6).

3.2.3.3.2 t-distributed stochastic neighbour embedding (t-SNE) visualisation

To evaluate the visual effectiveness of the Doc2Vec patent embeddings we use the t-SNE[2,3,4] visualisation method (Van Der Maaten & Hinton, 2008).

We use the t-SNE approach to form a visual representation of the patent text sections Doc2Vec vector relative to the categoric input feature determinants (3.2.2.2), which include the IPC primary section, IPC primary class, CPC primary section, CPC primary subsection and CPC primary group (Linderman & Steinerberger, 2019; Pezzotti et al., 2020; Van Der

[1]Tokenisation is the process of chopping the text into words or sentences, called tokens. At the same time, this removes punctuation and makes all elements lowercase (Brownlee, 2017c; Fonseca, 2019). Tokens are sequences of characters with context, grouped together as a useful semantic unit for processing (Manning et al., 2008). Stopwords, i.e. 'the' or 'a', are the most common words in any language, and is standard practice to remove them, because they are considered noise (Mattyws Grawe et al., 2017). However, since the model is also a contextual model, and given the complex patent language due to patent drafting (Dirnberger, 2011; Marco et al., 2019; WIPO, 2010), these words are important syntactical elements to understand the contextual meaning of words, sentences and paragraphs, and we chose to keep them (Hess, 2008; Marco et al., 2019). Stopwords are also referred to non-content bearing words, which do not impact knowledge of the topic area but impact the contextual meaning once removed (Agarwal & Yu, 2009; Trippe, 2015).

[2]The t-SNE is a non-linear dimensionality reduction algorithm for high-dimensional embedding visualization in a low-dimensional space (Van Der Maaten, 2009). It constructs a probability distribution over pairs of high-dimensional objects (Van Der Maaten, 2020). It then defines a similar probability distribution over the points in the low-dimensional map, and it minimizes the Kullback–Leibler divergence (KL divergence) between the two distributions with respect to the locations of the points in the map (Van Der Maaten, 2010), by gradient descent. It succeeds in preserving both global and local data structures that makes clusters visible at several scales (Skripnikova, 2019). It is also important to note that when minimising high dimensional space to a low dimension space, some information would not be recovered (Arora et al., 2018; Wattenberg et al., 2016).

[3]The Kullback–Leibler divergence (KL divergence) is a measure of how a probability distribution is different from reference probability distribution. Minimising the KL divergence, minimises the error of the representation.

[4]Gradient descent is an iterative optimization algorithm that aims to find a minimum of a function by taking small steps in the direction of the steepest decline.

Maaten & Hinton, 2008; Wattenberg et al., 2016; Whitehead et al., 2017)[1]. Similarly, Kim et al. (2020) propose a method for applying embedded feature vectors to identify text features in patent documents by visualising them using t-SNE and evaluate their effectiveness. We choose the t-SNE optimised hyperparameters based on previous research[2] (Van Der Maaten & Hinton, 2008).

Table 3.4[3] summarises the main observations from the t-SNE visualisations of the Doc2Vec vector of the patent text sections, shown in Fig. 3.7-3.10. We observe some evidence that suggest that our Doc2Vec vector inherits the semantic and syntactic properties of the words and sentences (Le et al., 2014). This ensures that the context of the document is maintained, in addition to the global context of the corpus (Dai et al., 2015).

Table 3.4 Observations from the t-SNE algorithm visualisations of the Doc2Vec vector for the patent text sections of abstract, claims, summary, title, for the Fig. 3.7-3.10

Section[a,b,c]	Abstract	Claims	Summary	Title
Fig.	Fig. 3.7	Fig. 3.8	Fig. 3.9	Fig. 3.10
Categoric input feature determinants (3.2.2.2)	Limited differences between perplexities 30 and 50 in the overall shape of the visualisation for all input features (2, 3). There is some cluster formation, which is better observable for IPC primary section and perplexity 50 (3). The IPC primary section and IPC primary class have a similar shape (2, 3, 5, 6), and the CPC primary section, CPC primary subsection and CPC primary group, indicating that the overall language found within the hierarchical levels is similar (8, 9, 11, 12, 14, 15). With increasing granularity, i.e. down the hierarchy levels of the technological classifications, CPC primary subsection, CPC primary group, we observe that the clusters are not well formed (12, 14, 15). This can be attributed to two reasons: (i) the training of the t-SNE algorithm is insufficient, and requires more time to converge; (ii) the language becomes more overlapping in a similar context, and the proximity of the points increases.	The IPC primary section and IPC primary class have distinct clusters from perplexity=5 (1, 4). This is expected because of the claim length, and could possibly indicate that the contextual similarity is higher. With increasing granularity, i.e. down the hierarchy levels of the technological classifications, we can observe that the language becomes more overlapping because there is limited clustering. However, this limited clustering unlike Fig. 3.7, could be partly because of two reasons: (i) the language becoming more repetitive because of broad claims, (ii) the language within the same fields becoming more specific and similar (12, 14, 15).	The clusters are more well formed than all the other patent text sections, especially for perplexity=30, and even with increasing granularity (2, 5, 8, 11, 14). Patent summaries contain both descriptive and technical language, balancing semantic and syntactic context, which could be partly because of this.	There is very limited cluster formation, partly because the title consists of only a few words, which possibly cannot fully represent the contextual and semantic relationships.

[a]The aim of this analysis is to visually represent the patent text sections, with the adapted Doc2Vec vectors. The colours represent the technological classes, i.e. points with colour grey are within IPC primary section H. For simplicity, due to the large number of classes, subsections and groups, we do not explicitly mention the technological classes in the descriptive text or the figures.

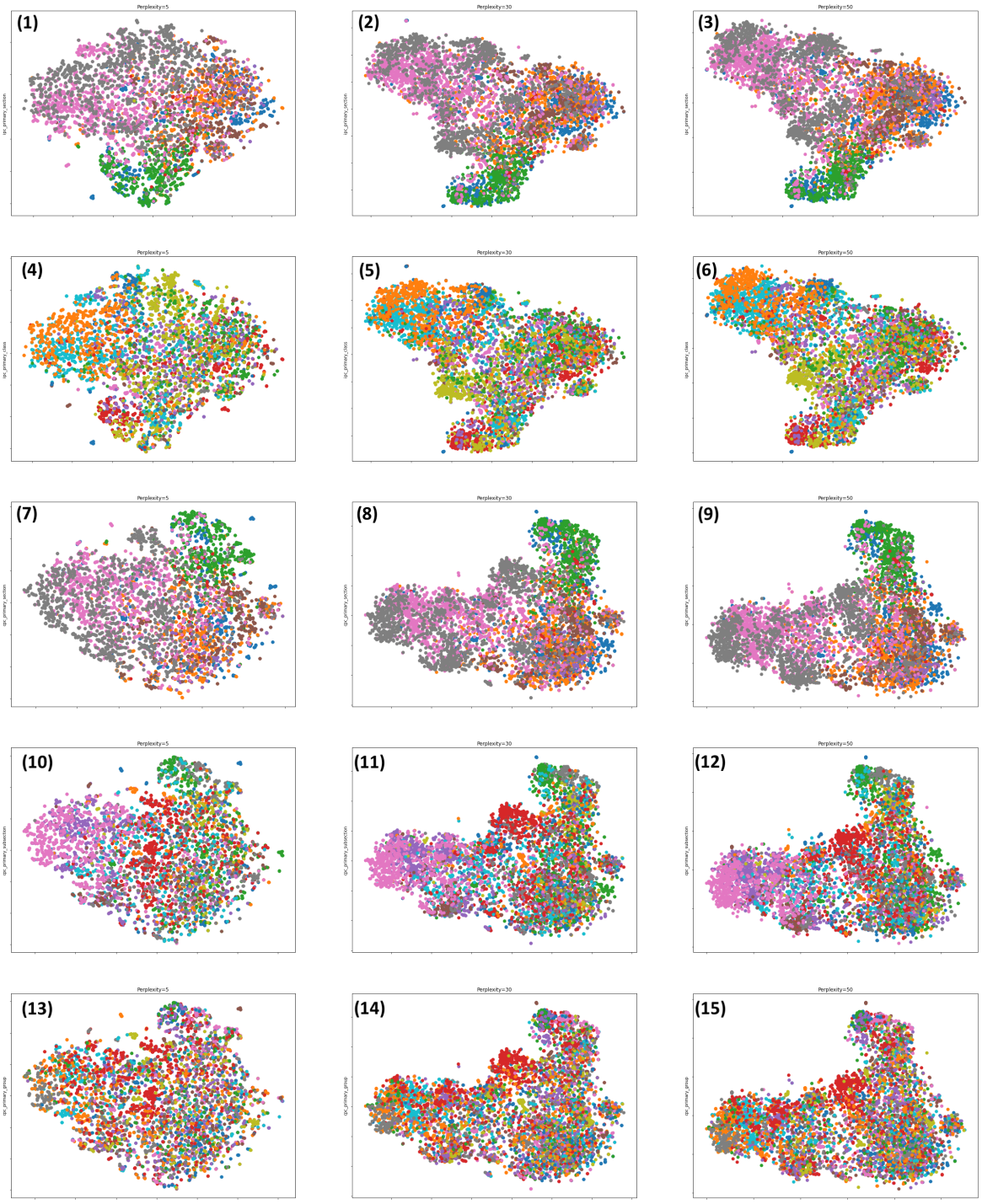
[b]The associated numbers in brackets refer to the numbers in Fig. 3.7-3.10.

[c]The table provides an overview and some main observations from the t-SNE algorithm visualisations.

[1]We also compile the t-SNE for the patent text sections Doc2Vec vector relative to the categoric output/ target feature proxies (3.2.2.3). However, for simplicity, we do not report it here.

[2]For all the experiments, we use an iteration cycle (epochs) of 1000, a learning rate of 200, and a set of perplexities of 5, 30, and 50 (Belkina et al., 2019; Cao & Wang, 2017). Perplexity is the balance between local and global aspect of the data, i.e. is the number of neighbours each point has, which has a complex effect on the resulting visualisations, with typical values ranging between 5 and 50 (Wattenberg et al., 2016).

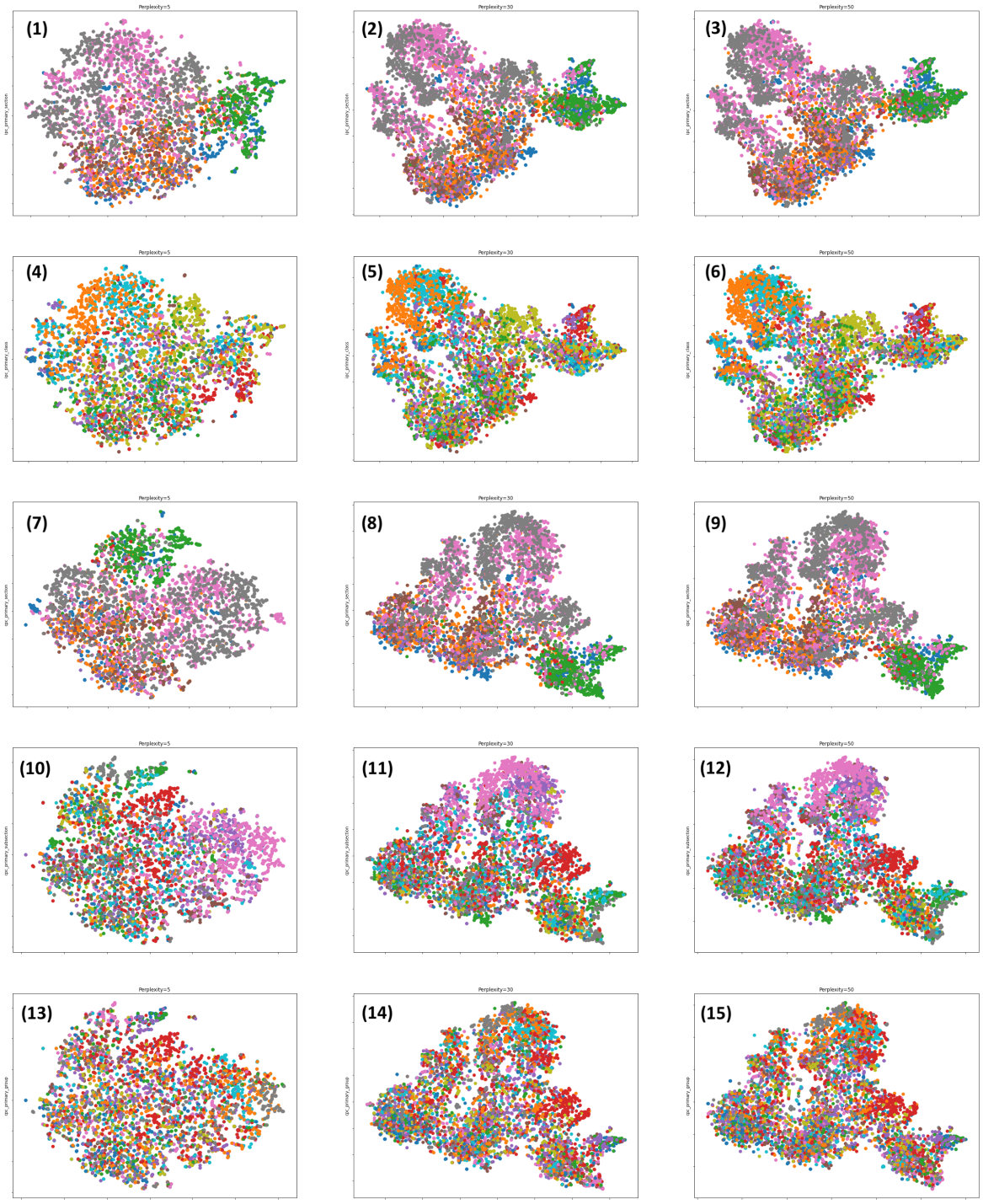
[3]The numbers in brackets in Table 3.4 refer to Fig. 3.7-3.10, e.g. Fig. 3.7[2] refers to the abstract and the analysis on IPC primary section vs. pervar(30).



[a]Categoric input feature determinants appear in order: IPC primary section, IPC primary class, CPC primary section, CPC primary subsection, CPC primary group (vertical axis), with perplexity variation (pervar) of $[5, 30, 50]$ (horizontal axis).

[b]The associated numbers represent the following combinations: (1) IPC primary section vs. pervar(5), (2) IPC primary section vs. pervar(30), (3) IPC primary section vs. pervar(50), (4) IPC primary class vs. pervar(5), (5) IPC primary class vs. pervar(30), (6) IPC primary class vs. pervar(50), (7) CPC primary section vs. pervar(5), (8) CPC primary section vs. pervar(30), (9) CPC primary section vs. pervar(50), (10) CPC primary subsection vs. pervar(5), (11) CPC primary subsection vs. pervar(30), (12) CPC primary subsection vs. pervar(50), (13) CPC primary group vs. pervar(5), (14) CPC primary group vs. pervar(30), (15) CPC primary group vs. pervar(50).

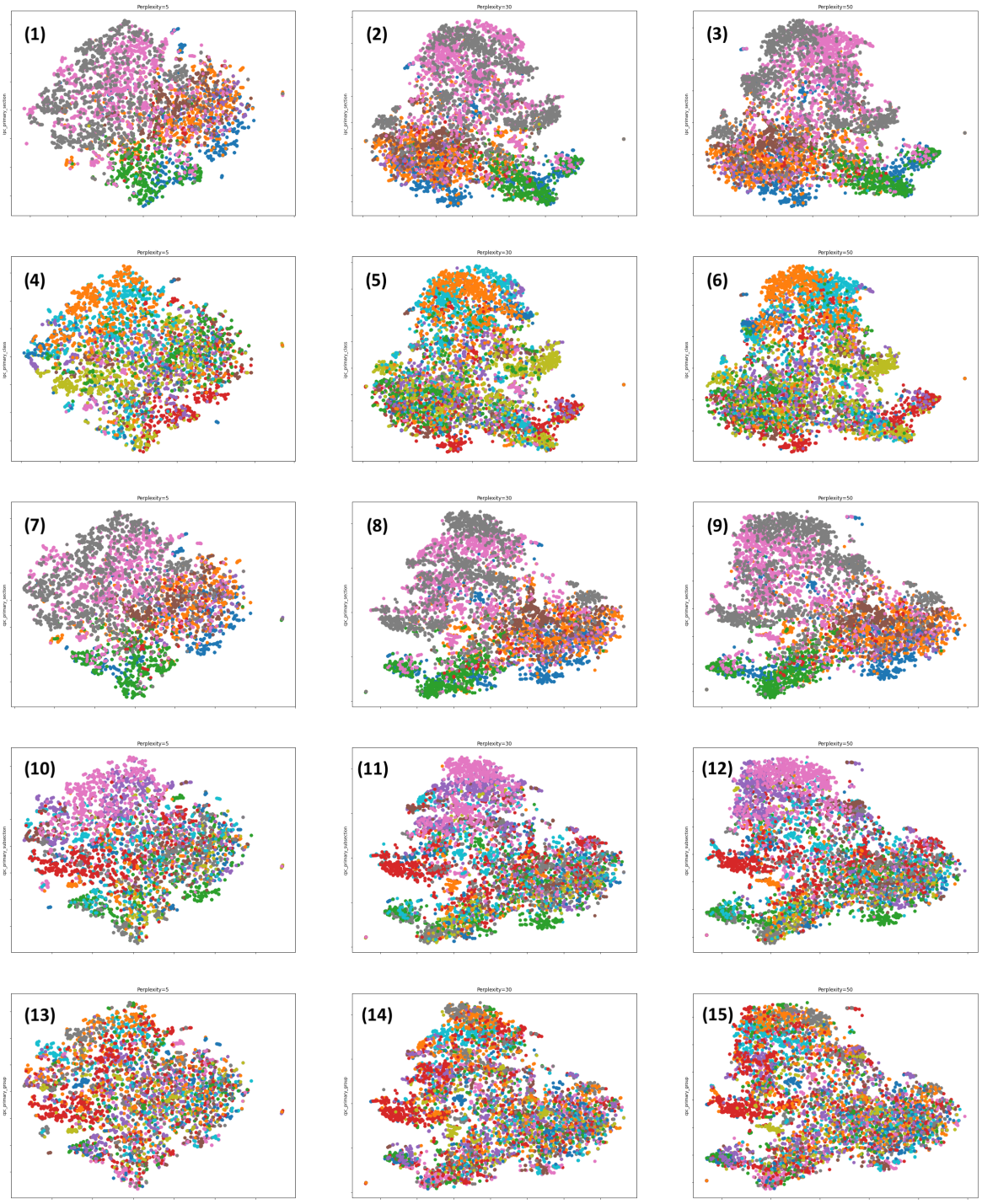
Fig. 3.7 t-SNE visualisation for abstract vs. categoric input feature determinants[a,b]



[a]Categoric input feature determinants appear in order: IPC primary section, IPC primary class, CPC primary section, CPC primary subsection, CPC primary group (vertical axis), with perplexity variation (pervar) of $[5, 30, 50]$ (horizontal axis).

[b]The associated numbers represent the following combinations: (1) IPC primary section vs. pervar(5), (2) IPC primary section vs. pervar(30), (3) IPC primary section vs. pervar(50), (4) IPC primary class vs. pervar(5), (5) IPC primary class vs. pervar(30), (6) IPC primary class vs. pervar(50), (7) CPC primary section vs. pervar(5), (8) CPC primary section vs. pervar(30), (9) CPC primary section vs. pervar(50), (10) CPC primary subsection vs. pervar(5), (11) CPC primary subsection vs. pervar(30), (12) CPC primary subsection vs. pervar(50), (13) CPC primary group vs. pervar(5), (14) CPC primary group vs. pervar(30), (15) CPC primary group vs. pervar(50).

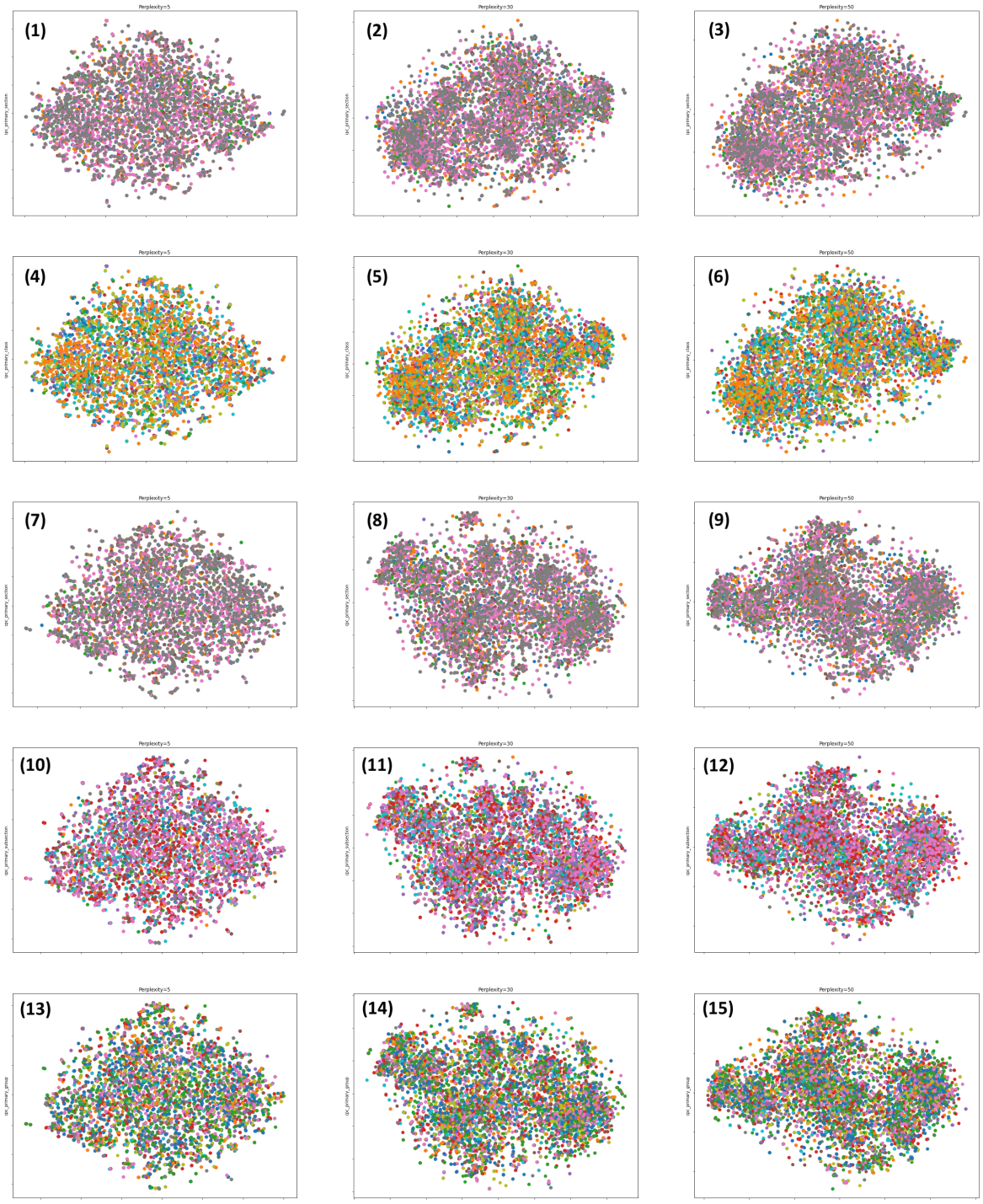
Fig. 3.8 t-SNE visualisation for claims vs. categoric input feature determinants[a,b]



[a]Categoric input feature determinants appear in order: IPC primary section, IPC primary class, CPC primary section, CPC primary subsection, CPC primary group (vertical axis), with perplexity variation (pervar) of $[5, 30, 50]$ (horizontal axis).

[b]The associated numbers represent the following combinations: (1) IPC primary section vs. pervar(5), (2) IPC primary section vs. pervar(30), (3) IPC primary section vs. pervar(50), (4) IPC primary class vs. pervar(5), (5) IPC primary class vs. pervar(30), (6) IPC primary class vs. pervar(50), (7) CPC primary section vs. pervar(5), (8) CPC primary section vs. pervar(30), (9) CPC primary section vs. pervar(50), (10) CPC primary subsection vs. pervar(5), (11) CPC primary subsection vs. pervar(30), (12) CPC primary subsection vs. pervar(50), (13) CPC primary group vs. pervar(5), (14) CPC primary group vs. pervar(30), (15) CPC primary group vs. pervar(50).

Fig. 3.9 t-SNE visualisation for summary vs. categoric input feature determinants[a,b]



[a]Categoric input feature determinants appear in order: IPC primary section, IPC primary class, CPC primary section, CPC primary subsection, CPC primary group (vertical axis), with perplexity variation (pervar) of $[5, 30, 50]$ (horizontal axis).

[b]The associated numbers represent the following combinations: (1) IPC primary section vs. pervar(5), (2) IPC primary section vs. pervar(30), (3) IPC primary section vs. pervar(50), (4) IPC primary class vs. pervar(5), (5) IPC primary class vs. pervar(30), (6) IPC primary class vs. pervar(50), (7) CPC primary section vs. pervar(5), (8) CPC primary section vs. pervar(30), (9) CPC primary section vs. pervar(50), (10) CPC primary subsection vs. pervar(5), (11) CPC primary subsection vs. pervar(30), (12) CPC primary subsection vs. pervar(50), (13) CPC primary group vs. pervar(5), (14) CPC primary group vs. pervar(30), (15) CPC primary group vs. pervar(50).

Fig. 3.10 t-SNE visualisation for title vs. categoric input feature determinants[a,b]

3.3 Exploratory data analysis (EDA)

We perform an exploratory data analysis (EDA), following the data preparation (transformation) process (see 3.2), for our full dataset (100FD), as to confirm results from previous studies to validate our dataset and to identify relationships between the data. Firstly, we produce the patent distributions per year and per technological area in terms of IPC and CPC classifications, and then the descriptive statistics for the full dataset[1,2].

Fig. 3.11 shows the distribution of the full dataset (100FD) by publication year, CPC classification section and IPC classification section. We observe that our dataset's distributions follow the USPTO distributions[3] for granted patents (WIPO, 2019a,c), with an exponential increase in the number of patents in the last two decades (50% of the patents have been granted after 2003) (Harhoff et al., 2007; WIPO, 2020). The CPC section of G (physics) has the highest number of patents, followed by H (electricity) and B (performing operations/ transporting), where as D (textiles/ paper) has the lowest number of patents (EPO, 2017), which follows a similar trend to the IPC classification section.

Table 3.5 shows the descriptive statistics for the numeric input feature determinants (3.2.1) and categoric output/ target feature proxies (3.2.2.2) for the full dataset (100FD)[4]. We observe

[1]In part, we perform a sensitivity analysis on the distributions of the dataset (chapter 3) to explore the multi-dimensional input space (Leamer, 1985). We produce scatter plots and perform a correlation analysis. We identify very weak correlations, with correlation coefficients between $+/-0.15$ (Saltelli et al., 2008, 2004; Zhou & Lin, 2008). For simplicity, due to the large number of the scatter plots and the large matrix produced from the correlation analysis, we do not include them in the thesis. A limitation of our sensitivity analysis in relation to the dataset is that we could have focus more on the quantification of the uncertainty in each input (numeric, text, categoric) relative to the categoric outputs, and perform more in depth one-at-time (OAT) analysis, local derivative analysis (partial derivative of the output with respect to an input), and a more detailed variance-based analysis (James et al., 2013; Saltelli et al., 2008, 2004). However, this is partially counterbalanced by the focus on developing a deep learning methodology (see chapter 4) using artificial neural networks, which are highly adaptive, with a high noise tolerance (Bishop, 2006). Thus, the information gain from collinear data outperforms the noise concerns (Murphy, 2012).

[2]Driven by the artificial intelligence and machine learning literature, we perform a sensitivity analysis on the model development (see chapter 4) (Leamer, 1985; Saltelli et al., 2008, 2004). We perform an in-depth network optimisation using a quasi-experimental approach to identify the optimal hyperparameters and model parameters (see 4.4) (Brownlee, 2019e; Murphy, 2012). This follows the emulator methodology for evaluating the robustness and reliability of our proposed approach (see 4.5) (Becker et al., 2013; Gramacy & Taddy, 2010; Oakley & O'Hagan, 2004; Sudret, 2008; Wang et al., 2019). We develop an in-depth evaluation strategy for training, validation and testing, which also includes cross validation and ensemble learning (see 4.5.2). A limitation of our sensitivity analysis in relation to the model development, is that we could have concentrated on using more advance ensemble learning models for varying the inputs and evaluating the impact on the output, and also focus on the model simplification (i.e. identifying and removing redundant parts of the model input and structure).

[3]Lens.org open datasets - USPTO full granted patents from 1976-2019, URL: https://link.lens.org/mf0zpKCXDzh.

[4]The dataset consists of the full USPTO granted patents from 1976-2019 (see 3.1.2), and the model development (see chapter 4) considers all the population. In principle, focusing only on granted patents and not including patent applications can introduce survival bias (Brown et al., 1992; Deng et al., 1999; Groeneveld &

that backward citations and forward citations (citations_t4, citations_t8, citations_t12) are positively skewed (Hall, 2005)[1].

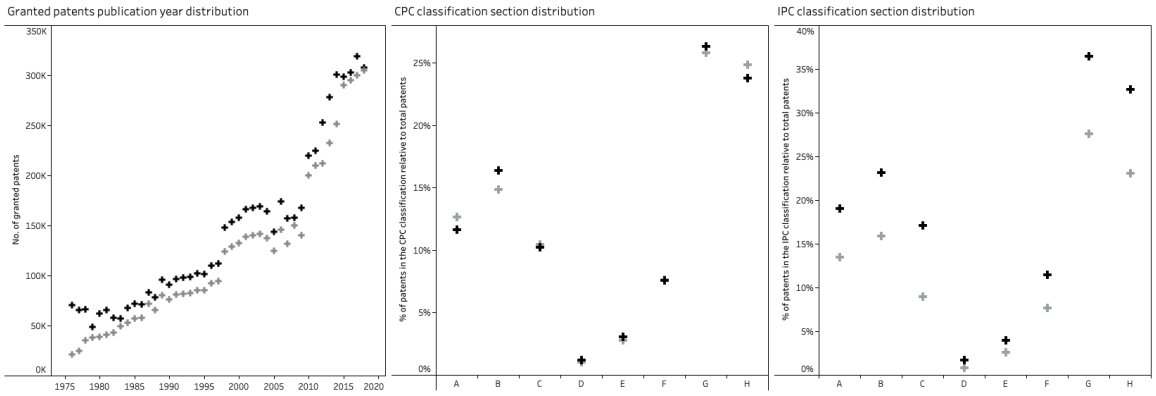


Fig. 3.11 Distribution of granted patents by (a) publication year (b) CPC classification section (c) IPC classification section (legend: black colour = USPTO full dataset, grey colour = our developed dataset)

The originality index is high, which is expected because a patent is granted based on novelty and inventiveness (Squicciarini et al., 2013). The mean filing year is 2001, with 50% of the patents having been filed until 2003. On average, patent family size is 3.98, with patents with more than 5 patent family members being above the 75th percentile of the distribution. Patent scope is positively skewed, with a long tail (Marco et al., 2019). In all categoric output feature proxies[2].

Meeden, 1984; Lin & Chen, 2005; Seru, 2014); however, granted patents provide exclusive market rights to the firms for exploitation, which gives an indication of the market value of patented inventions (Hall, 2005). We develop an in-depth evaluation strategy with variations of the dataset (see 4.5.1). Given that the number of datapoints is restricted to the patent reaching age 4, 8 and 12, survival bias can be introduced given that the patents that have not reached the required age are excluded (Brown et al., 1992; Groeneveld & Meeden, 1984). However, all patents remaining are treated similarly, with the model taking into consideration high value, V_H and low value, V_L patents at the same point in time (Murphy, 2012). This was introduced to ensure that there is a complete set of input feature determinants and output proxies for every patents (see 4.5). This allows for inter-model comparison, i.e. within the same variant of the dataset, and intra-model comparison, i.e. across the variants of the dataset (see Fig. 4.12 and 4.13).

[1]From Table 3.5, it is evident that all distributions are positively skewed, with a Pearson's first skewness coefficient (mode skewness) of 0.24 for citation_t4, 0.26 for citation_t8 and 0.20 for citation_t12. In addition, the Yule's coefficient of skewness is 0.33 for citation_t4, 0.33 for citation_t8 and 0.40 for citation_t12. The positive skewed distribution follows the distribution identified by Hall (2005), which also justifies the use of a cost-sensitive function, i.e. the focal loss, to account for imbalanced datasets (see 4.4.2.2). A limitation is that we could focus more on variance-based methods, which in turn could improve our loss function training.

[2]From Table 3.5, the renewals (t4, t8, t12) distribution is negatively skewed, with the Pearson's first skewness coefficient (mode skewness) of -0.15, and the Yule's coefficient of skewness of -0.11. This is similar to the distribution by Lanjouw et al. (1998); Squicciarini et al. (2013), with the majority of patents being renewed by firms in the fear of losing out. It is also evident that the generality distributions are positively skewed, with the

Table 3.5 Descriptive statistics for numeric input feature determinants and categoric output/ target feature proxies for the full dataset (100FD)

Features[a,b]	Variable name[c]	Datapoints [d]	Mean	Standard Deviation	Min.	Percentiles 25p	50p	75p	Max.
Numeric input feature determinants (see 2.1.2.4, Table 2.5, and 3.2.1)	Backward citations	5190340	19.33	47.51	0.00	6.00	11.00	19.00	5277.00
	Classification - many field	5190340	0.43	0.49	0.00	0.00	0.00	1.00	1.00
	Dependent claims	5190340	15.53	11.46	1.00	8.00	14.00	20.00	-
	Family size	5190340	3.98	4.07	1.00	1.00	3.00	5.00	57.00
	Filing year	5190340	2001.18	10.64	1976.00	1994.00	2003.00	2010.00	2018.00
	Independent claims	5190340	7.80	9.85	1.00	2.00	4.00	10.00	-
	Non-patent literature citations	5190340	4.40	16.85	0.00	0.00	0.00	2.00	2356.00
	Num_of_unique_cpc_additional_group	5190340	0.81	0.96	0.00	0.00	1.00	1.00	17
	Num_of_unique_cpc_additional_section	5190340	0.71	0.75	0.00	0.00	1.00	1.00	8
	Num_of_unique_cpc_additional_subgroup	5190340	2.19	4.58	0.00	0.00	1.00	3.00	265
	Num_of_unique_cpc_additional_subsection	5190340	0.74	0.83	0.00	0.00	1.00	1.00	13
	Num_of_unique_cpc_invention_group	5190340	1.58	0.91	0.00	1.00	1.00	2.00	24
	Num_of_unique_cpc_invention_section	5190340	1.26	0.50	0.00	1.00	1.00	1.00	6
	Num_of_unique_cpc_invention_subgroup	5190340	3.63	3.75	0.00	1.00	3.00	4.00	241
	Num_of_unique_cpc_invention_subsection	5190340	1.41	0.70	0.00	1.00	1.00	2.00	15
	Originality index	5190340	0.73	0.21	0.00	0.66	0.80	0.88	1.00
	Patent scope	5190340	1.92	1.20	1.00	1.00	2.00	2.00	44.00
	Radicalness index	5190340	0.37	0.27	0.00	0.15	0.34	0.57	1.00
	Total claims	5190340	15.58	11.62	1.00	8.00	14.00	20.00	887.00
Categoric output/ target feature proxies (see 2.1.2.3, Table 2.4, and 3.2.2.2)	Citation_t4	4192600	2.16	4.83	0.00	0.00	1.00	3.00	509.00
	Citation_t8	3226554	6.27	12.63	0.00	1.00	3.00	7.00	2702.00
	Citation_t12	2560696	11.87	39.64	0.00	1.00	4.00	11.00	3981.00
	Generality	5190340	0.47	0.28	0.00	0.28	0.52	0.69	1.00
	Generality_t4	4192600	0.14	0.25	0.00	0.00	0.00	0.26	1.00
	Generality_t8	3226554	0.25	0.30	0.00	0.00	0.00	0.50	1.00
	Generality_t12	2560696	0.29	0.31	0.00	0.00	0.20	0.59	1.00
	Grant_Lag	5190340	984.41	564.82	0.00	593.00	844.00	1233.00	14060.00
	Renewals (t4, t8, t12)	-	8.32	4.59	0.00	4.00	9.00	13.00	39.00
	Quality_index_4	5190340	0.27	0.13	0.01	0.17	0.26	0.34	0.99

[a]The descriptive statistics are calculated on the full dataset (100FD). The total number of datapoints, n, for the full dataset (100FD) are 5190340 granted patents between 01.01.1976 - 01.01.2019 (see 3.1.1).

[b]Table 4.1 includes the operationalisation of all features (proxies and determinants) for deep learning, including the variable name, type, dimension and operational definition.

[c]Variable names are arranged in alphabetical order and are calculated on the patent level (see 2.1.2).

[d]Some of the categoric output/ target proxies have different total number of datapoints (i.e. total number of complete variables per patent), which depends on the time T, where $T = 4, 8, 12$ in T years after the grant date.

Pearson's first skewness coefficient (mode skewness) of 0.56 for generality_t4, 0.83 for generality_t8 and 0.30 for generality_t12. In addition, the Yule's coefficient of skewness is 1.00 for generality_t4, 1.00 for generality_t8 and 0.33 for generality_t12. The relative increase in forward citations is less than the diversification in the IPC classes, showing that there is a small number of patents covering a diverse number of fields, with increasing diversification as the number of forward citations increases (i.e. forwarded cited patents are filed in a diverse number of fields). This is then followed by a sudden drop in positive skewness, showing a concentration of the forward citations in the same number of fields. The patent quality index 4, as defined by Squicciarini et al. (2013), is close to an asymmetric distribution, with the Pearson's first skewness coefficient (mode skewness) of 0.05, and the Yule's coefficient of skewness of -0.02. This is partly the reason for the improved model performance in Table 5.15.

Chapter 4

Developing the deep learning algorithmic approach

In this chapter, we describe and explain the development of the deep learning (DL)[1] algorithmic approach (Schmidhuber, 2015), an artificial intelligence (AI) methodology, for the valuation of patented invention, i.e. patents[2]. We aim to show the design and development of the algorithm, including the tests performed to arrive at optimised prediction models[3]. Fig. 4.1[4] shows the process flowchart for the development of the deep learning approach, which is supported by computational resources[5] (see 1.2 and Fig. 1.1).

We follow a forecasting approach to structure the problem into a supervised learning paradigm[6] and a classification task (4.1). We describe and explain the process of developing the network architecture (4.2)[7].

[1]For the purpose of this research (see 2.2), we use the term *deep learning (DL)* to describe artificial neural networks (ANN), in supervised learning paradigms, defined by the depth of the credit assignment paths, which are chains of possibly learnable, causal links between inputs and outputs (Hinton et al., 2006), i.e. finding weights that make the neural network exhibit desired behaviour (Schmidhuber, 2015). These are also known as deep (and wide) neural networks (Cheng et al., 2017; Goodfellow et al., 2016; Shaked et al., 2016).

[2]From previous studies, the application of artificial intelligence (AI) methodologies for patent valuation is limited, as identified in chapter 2.

[3]The code (Aristodemou, 2020a) is written in Python language (Van Rossum & Drake, 1995), and uses the libraries of Tensorflow (Abadi et al., 2016a) and Keras (Chollet & Others, 2015). Tensorflow (Abadi et al., 2016b), URL: https://tensorflow.org. Keras (Chollet & Others, 2015), URL: https://keras.io.

[4]Fig. 4.1 is a subset of Fig. 1.1.

[5]The data is stored in the cloud and processed with virtual machines using Microsoft Azure (Microsoft, 2020) and Google AI Platform servers (Google, 2020a). The code is written in Python (Van Rossum & Drake, 1995), and is stored and maintained on GitHub (Github, 2020).

[6]Supervised learning is when a learning task infers a function from the analysis of the training data, given a set of mapped input-output pairs, and can determine the mapping of new examples (Bishop, 2006; Goodfellow et al., 2016) (see 2.2.2).

[7]The network architecture is determined from the analysis of previous literature (see 2.1.2.6, 2.2.2.2 and 2.2.2.3.2). We focus on deep and wide artificial neural networks, i.e. deep learning, given the limited applications

We take a quasi-experimental network optimisation approach (4.4), for the optimisation, i.e. the error reduction of the error function between target output value and predicted output value, of the network architecture (4.2). This consists of two main optimisation tasks: the optimisation of the parameters (4.4.1), and the optimisation of the gradient descent algorithm (4.4.2). We evaluate the error function derivative of these optimisations with the evaluation metrics: accuracy, confusion matrix, F1-score, false negative rate (FNR), log loss, mean absolute error (MAE), precision, and recall (4.3). Section 4.5 describes the implementation tests of the deep and wide neural network, with emphasis on the dataset split (4.5.1), and evaluation strategies (4.5.2).

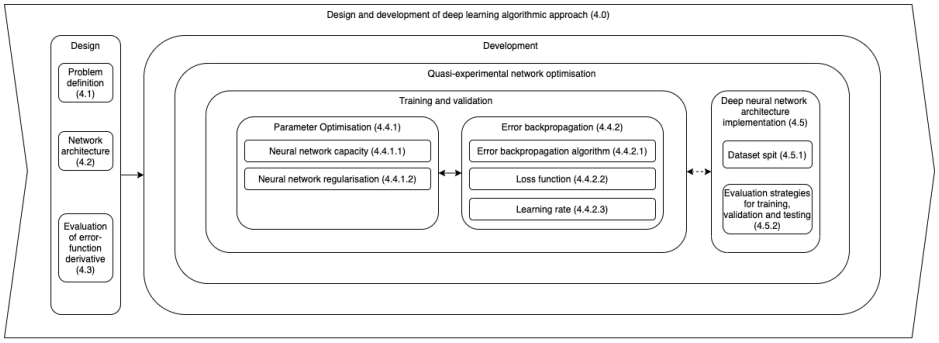


Fig. 4.1 Process flowchart of the proposed deep learning approach (a subset of Fig. 1.1)

4.1 Supervised learning approach and problem definition

The analysis of the literature in chapter 2, reveals that for the analysis of patent data with artificial intelligence (AI) methodologies, the majority of the articles are found to be supervised learning paradigms (2.2.2.1). The most widely researched and implemented methodology is the multi-layer perceptron (MLP) artificial neural network (ANN) (Basheer & Hajmeer, 2000), with the implementation of the backpropagation (BP) learning algorithm (Schmidhuber, 2015). This applies for the analysis of patent data (see 2.2.2.2), with scholars mainly using the MLP ANN for technology classification (see Table 2.10).

However, there are limited studies of applying AI methods for valuing patents (see 2.1.2.6). From these, only a limited number make use of ANN, with low capacity[1] shallow

of this network architecture in previous studies (2.2.2.3.2), and most specifically artificial intelligence (AI) methodologies for patent valuation (2.1.2.6).

[1]The capacity of a neural network is defined as configuration of neurons or nodes and layers, i.e. the number of layers, the number of input nodes, the number of output nodes, and the number of nodes in each layer (Hopfield, 1982; Jia et al., 2016), and controls the scope of the types of mapping functions that it is able to learn (Brownlee, 2019g).

ANN[1], with binary classification, limited number of numeric and categoric features, and relatively small datasets. Thus, there is limited studies using deep and wide[2] ANN, i.e. deep learning[3], for the valuation of patents (Table 2.12).

The value of technologies can be modelled as the value of patents (Squicciarini et al., 2013). To measure patent value[4,5,6], we operationalise determinants[7] (see Table 2.5), to predict proxies[8] (see Table 2.4) (Frietsch et al., 2010; Reitzig, 2004). Most empirical implementations of patent value take the generic form in Eqn. 4.1a, which is operationalised by Eqn. 4.1b.

$$V_i = f(v_{i,j}) = g(C_{i,j,p_k}) \tag{4.1a}$$

$$C_{i,j,p_k} = h(C_{i,j,d_m}) \tag{4.1b}$$

where V_i is the value of patented invention i, $v_{i,j}$ is the value of patent j of patented invention i, C_{i,j,p_k} is a set of k patent value proxies p_k for every patent j, and C_{i,j,d_m} is a set of m patent value determinants d_m for every patent j.

[1]Deep neural networks are defined as networks with architectures with multiple hidden layers, where as shallow neural networks have one or two hidden layer (Delalleau & Bengio, 2011; Goodfellow et al., 2016; Murphy, 2012).

[2]The number of layer nodes is referred to as the width, and the number of layers is referred to as the depth, of the neural network (Abood & Feltenberger, 2018; Brownlee, 2019g).

[3]Brownlee (2019m), URL: https://machinelearningmastery.com/what-is-deep-learning/.

[4]The value of patents is rarely observable (Harhoff et al., 2003), and inductive approaches are more suitable to determine it (Reitzig, 2003).

[5]It has been argued in the literature that the nature of value recognition has an intrinsic and extrinsic dimension. The intrinsic dimension is derived from the intrinsic technological significance, and is represented by all that appear in the patent document, where as the extrinsic value is the potential to develop the market (Grimaldi & Cricelli, 2019). Moreover, it has also been suggested that the value of patents consists of: (i) the value of the patent rights, and (ii) the value of the invention (Jensen et al., 2011), with the former representing the patent premium, i.e. the value related to the market protection (Arora et al., 2008).

[6]The PatVal study has estimates of patent value (Gambardella et al., 2007, 2005). Given the difficulty in sourcing this dataset, we focus on alternative proxies capable of estimating patent value, of which some correlate with the patent premium (Jensen et al., 2011).

[7]We define patent value determinants to represent patent characteristic that have been used mainly as value determinants, correlated with patent value. They represent patent characteristics that have been used in the literature as explanatory variables and can be classified as ex-ante indicators. An ex-ante indicator is a patent characteristic that is related to the nature of a patented invention, and is defined immediately at the point or just after the patent is filed (Arts et al., 2013; Lee et al., 2018; Noh & Lee, 2020).

[8]We define patent value proxies as measures that can be used to approximate the value of patented inventions. They represent patent characteristics that have been used mainly in the literature as dependent variables and can be classified as ex-post indicators (van Zeebroeck & van Pottelsberghe de la Potterie, 2011a). An ex-post indicator is patent characteristic that is related to the impact and value of a patented invention, which may change over time (Arts et al., 2013; Lee et al., 2018; Noh & Lee, 2020).

Table 4.1 Patent value output/ target feature proxies and input feature determinants, operationalised for deep learning

Features[a]	Patent characteristics[b,c]	Variable name[d]	Variable type[e]	Dim. $(D^T)^f$	Operational Definition[g]
Categoric target/ [h] output proxies (C_{i,j,p_k})	Forward Citations	Citations_t4	C	$[d_0]$	0 if greater than 21 citations, otherwise 1
		Citations_t8	C	$[d_0]$	0 if greater than 21 citations, otherwise 1
		Citations_t12	C	$[d_0]$	0 if greater than 21 citations, otherwise 1
	Generality Index	Generality_t4	C	$[d_0]$	0 if greater than 0.75, otherwise 1
		Generality_t8	C	$[d_0]$	0 if greater than 0.75, otherwise 1
		Generality_t12	C	$[d_0]$	0 if greater 0.75, otherwise 1
	Grant Lag	Grant_Lag	C	$[d_0]$	0 if less than 600 days, otherwise 1
	Renewals	Renewal_t4	C	$[d_0]$	0 if renewal occurs in year 4 after grant date, otherwise 1
		Renewal_t8	C	$[d_0]$	0 if renewal occurs in year 8 after the grant date, given renewal_t4 occurred, otherwise 1
		Renewal_t12	C	$[d_0]$	0 if the renewal of the patent occurs in year 12 after the grant date, given renewal_t8 occurred, otherwise 1
	Quality Index[i]	Quality_Index_4	C	$[d_0]$	0 if greater than 0.45, otherwise 1
Input feature [j] determinants (C_{i,j,d_m})	Backward Citations	Backward citations	N	$[d_0]$	Number of backward citations
	Claims	Total claims	N	$[d_0]$	Number of claims
		Independent claims	N	$[d_0]$	Number of independent claims
		Dependent claims	N	$[d_0]$	Number of dependent claims
	Classification	CPC section	C	$[d_0,...,d_7]$	One hot encoding of the CPC section
		CPC subsection	C	$[d_0,...,d_{122}]$	One hot encoding of the CPC subsection
		CPC main group	C	$[d_0,...,d_{629}]$	One hot encoding of the CPC main group
		CPC invention section	N	$[d_0]$	Number of CPC invention sections
		CPC invention subsection	N	$[d_0]$	Number of CPC invention subsections
		CPC invention main group	N	$[d_0]$	Number of CPC invention main groups
		CPC invention subgroup	N	$[d_0]$	Number of CPC invention subgroups
		CPC additional section	N	$[d_0]$	Number of CPC additional sections
		CPC additional subsection	N	$[d_0]$	Number of CPC additional subsections
		CPC additional main group	N	$[d_0]$	Number of CPC additional main groups
		CPC additional subgroup	N	$[d_0]$	Number of CPC additional subgroups
		IPC section	C	$[d_0,...,d_7]$	One hot encoding of the IPC section
		IPC class	C	$[d_0,...,d_{130}]$	One hot encoding of the IPC class
		Many fields	N	$[d_0]$	1 if the patent is allocated to other fields
		Technology Field	C	$[d_0,...,d_{40}]$	Technology fields, IPC technology concordance (WIPO, 2009)
	Family	Family Size	N	$[d_0]$	Number of patent family members
	Filing year	Filing year	N	$[d_0]$	Filing year of the patent application
	Non-Patent Literature	NPL	N	$[d_0]$	Number of non-patent literature references
	Originality Index	Originality	N	$[d_0]$	Originality index
	Publication number	Pub. no. of granted patent	N	$[d_0]$	Publication Number of Granted Patent
	Radicalness Index	Radicalness Index	N	$[d_0]$	Radicalness index
	Scope	Patent scope	N	$[d_0]$	Number of unique IPC subclasses
	Text	Abstract	D2V	$[d_0,...,d_{299}]$	Doc2vec representation of the abstract
		Claims	D2V	$[d_0,...,d_{299}]$	Doc2vec representation of the claims
		Title	D2V	$[d_0,...,d_{299}]$	Doc2vec representation of the title
		Summary	D2V	$[d_0,...,d_{299}]$	Doc2vec representation of the summary

[a] Proxies are considered the output variables (equivalent to dependent variables and ex-post indicators), and determinants are considered the input features (equivalent to exploratory variables and ex-ante indicators) (Arts et al., 2013; Lee et al., 2018; Noh & Lee, 2020; van Zeebroeck & van Pottelsberghe de la Potterie, 2011a).

[b] Patent characteristics are calculated on the patent level, and are arranged in alphabetical order.

[c] Descriptive statistics for the numeric input feature determinants, and categoric output/ target feature proxies, and distributions of granted patents and technology classifications can be found in 3.3.

[d] Time T is chosen so that proxies are equivalent at the same point in time, where $T = 4, 8, 12$, i.e. in T years after the grant date

[e] The feature type: categoric (C), numeric (N), and text Doc2Vec representation (D2V) (see 3.2).

[f] The index of the vector dimension in computer science begins from 0.

[g] Value class definitions have been defined in 3.2.2.3 and Table 3.2. The number 0 represents high value patents, V_H, and the number 1 represents low value patents, V_L.

[h] Categoric target/ output proxies are identified, defined and described in 2.1.2.3, 2.1.2.5, and Tables 2.4 and 2.6. These have been transformed into categoric target/ output feature proxies in 3.2.2.3, and Table 3.2.

[i] It's a composite indicator, which follows the definition by Squicciarini et al. (2013) (Table 2.6).

[j] Input determinants are identified, defined and described in 2.1.2.4 and Table 2.5. Depending on their data type, i.e. numeric, categoric, text, these have been transformed into numeric input feature determinants in 3.2.1, categoric input feature determinants in 3.2.2.2, and Doc2Vec vector text input feature representation in 3.2.3.

Table 4.1 shows the patent value proxies belonging to set C_{i,j,p_k}, and the patent value determinants belonging to set C_{i,j,d_m}, that we use in our model development of using deep and wide ANNs (see 4.2). C_{i,j,p_k} includes all patent value proxies (see 2.1.2.3 and Table 2.4), and one composite index (see 2.1.2.5 and Table 2.6). These have been transformed into categoric target/ output feature proxies (see 3.2.2.3), with class definitions for a high value V_H and low value V_L patents[1] found in Table 3.2. C_{i,j,d_m} includes all patent value determinants (see 2.1.2.4 and Table 2.5). Depending on their data type, i.e. numeric, categoric, text, these have been transformed into numeric input feature determinants (3.2.1), categoric input feature determinants (3.2.2.2), and Doc2Vec vector text input feature representation (3.2.3).

4.2 Network architecture

Artificial neural networks (ANNs) are computational methodologies that can solve many complex real world problems[2] (Basheer & Hajmeer, 2000; Hagan et al., 1995). ANNs are modelled after biological neurons, with complex functions (Gupta, 2000; Murphy, 2012). They tend to outperform traditional methods, such as logistic regression methods, when the dimensionality and non-linearity of the problem increases, because they have a high noise tolerance, learning and generalisation capabilities (Basheer & Hajmeer, 2000; Hill et al., 1993; Lee et al., 1989; Sargent, 2001).

ANNs learn to approximate a function, which is being governed by a mathematical function, called the mapping function, and it is this function that a supervised learning paradigm algorithm seeks to best approximate[3]. ANNs seek to approximate the mapping function represented by the data observations. This is achieved by calculating the error between the predicted output variables, calculated by the model, and the expected target output variables, and minimizing this error during the training process (Bishop, 2006; Goodfellow et al., 2016). ANNs are known as universal adaptive approximators and in theory can be used to approximate any function (Murphy, 2012; Widrow & Lehr, 1993).

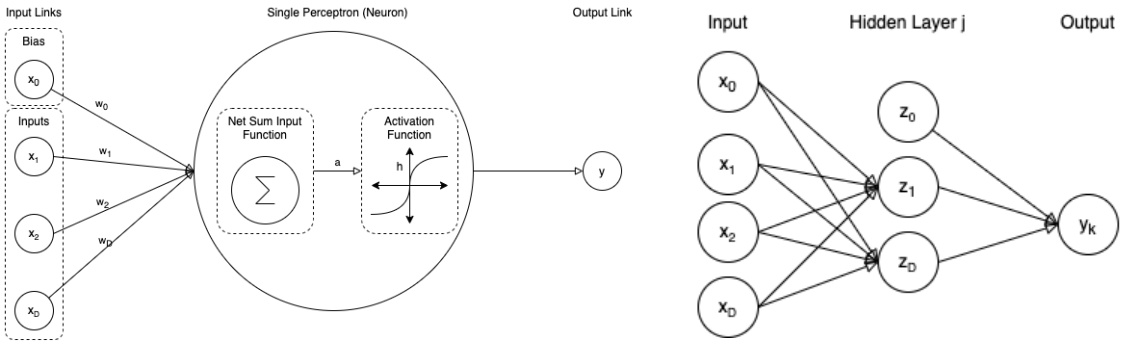
[1]The timeline and calculations of forward citations and the generality index, is aligned with the patent renewal timeline, i.e. T years after the grant date, where $T = \{4, 8, 12\}$ (Choi et al., 2020; Squicciarini et al., 2013).

[2]There are many complex real world problems that deploy artificial neural networks (ANNs) and deep learning (DL) (Baruffaldi et al., 2020; WIPO, 2019b). For example, De Fauw et al. (2018) apply a deep learning architecture to diagnose retinal disease and subsequent doctor referrals. Recently, Ozturk et al. (2020) also apply a neural network to detect the COVID-19 virus from patients' X-rays (Tietze et al., 2020b). There are also some limited studies that have deployed ANNs in the patent domain (Aristodemou & Tietze, 2018b). These have been reviewed in Table 2.9, and 2.2.2.2. For example, Trappey et al. (2006) classify a patent into its possible IPC classification, based on its prior art.

[3]Brownlee (2020e), URL: https://machinelearningmastery.com/neural-networks-are-function-approximators/.

Several ANN methodologies[1], have been developed over the years, with many network architectures, parameter and network optimisation techniques, and error function derivatives[2] (Hudson & Postma, 1982; Maren, 1991; Murphy, 2012; Schmidhuber, 2015). We use the multi layer perceptron (MLP) feed-forward ANNs, because it has the highest practical value and is the most widely researched and implemented methodology in other fields[3] (Basheer & Hajmeer, 2000; Bishop, 2006; Gupta, 2000; Murphy, 2012).

Neural networks have at least two physical components, the processing elements and the connections. The processing elements are called neurons, and the connections between the neurons are known as links. In the case of MLP, neurons are known as perceptrons (Rosenblatt, 1958). Every link has a weight parameter associated with it. Each neuron receives stimulus from the neighbouring neurons connected to it, processes the information, and produces an output. Neurons that receive stimuli from outside the network (i.e., not from neurons of the network) are called input neurons. Neurons on the output layer are called output neurons. Neurons that receive stimuli from other neurons and whose output is a stimulus for other neurons in the neural network are known as hidden neurons, and are part of the hidden layers (Gupta, 2000).



(a) (b)

Fig. 4.2 Network diagrams of 4.2a single perceptron (neuron) representation with 1 layer; 4.2b artificial neural network (ANN) with 2 layers

Fig. 4.2a shows the single perceptron, which is essentially a single layer neural network[4]. The perceptron has input links, a bias input term, a net input function, a non-linear activation function, and an output link, similarly to ANNs[5]. The perceptron computes a single output

[1]Van Veen (2016), URL: https://www.asimovinstitute.org/neural-network-zoo/.

[2]Some of these ANN methodologies have also been applied for the analysis of patent data (see 2.2.2.2 and Tables 2.9 and 2.10).

[3]This applies to the patent domain with limitations in the applications and applied methods (see 4.1).

[4]Sharma (2017b), URL: https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53.

[5]Chandra (2018), URL: https://towardsdatascience.com/perceptron-learning-algorithm-d5db0deab975.

from multiple real-valued inputs by forming a linear sum combination according to its input weights and then transfers the output through a non-linear activation function[1], to give an output link[2]. This is a functional transformation, which can be represented as follows:

$$a = \sum_{i=1}^{D} w_i x_i + w_0 x_0 \tag{4.2a}$$

$$y = h(a) = h\left(\sum_{i=1}^{D} w_i x_i + w_0 x_0\right) \tag{4.2b}$$

where $x = \{x_1, ..., x_D\}$ is a set of inputs, $w = \{w_1, ..., w_D\}$ is a set of weights (adjustable parameters during training), $h(.)$ is a non-linear activation function, $b = w_0 x_0$ is the bias parameter, a is an activation, and y is the output.

Neural networks use basis functions that follow the same form as the perceptron, so that each basis function is itself a non-linear function of a linear combination of the inputs, where the coefficients in the linear combination are adaptive parameters (Murphy, 2012). Fig. 4.2b shows a 2 layer[3] artificial neural network (ANN), with one input layer, one hidden layer and one output layer. The input, hidden and output variables are represented by nodes, with weight parameter connectors, and the bias parameters.

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} x_0 \tag{4.3a}$$

$$z_j = h(a_j) \tag{4.3b}$$

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)} z_0 \tag{4.3c}$$

$$y_k = \sigma(a_k) \tag{4.3d}$$

$$y_k(\mathbf{x}, \mathbf{w}) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} x_0\right) + w_{k0}^{(2)} z_0\right) \tag{4.3e}$$

$$y_k(\mathbf{x}, \mathbf{w}) = \sigma\left(\sum_{j=0}^{M} w_{kj}^{(2)} h\left(\sum_{i=0}^{D} w_{ji}^{(1)} x_i\right)\right) \tag{4.3f}$$

[1] Sharma (2017a), URL: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

[2] Honkela (2001), URL: https://users.ics.aalto.fi/ahonkela/dippa/node41.html.

[3] We adopt the terminology by Murphy (2012), where a 2 layer ANN is a network with 2 layers of adaptive weights, i.e. one hidden layer.

where $j = 1, ..., M$, and the subscript (1) indicates the corresponding parameters are in the first layer of the network, z_j are the hidden units, $h(.)$ is the activation function[1] of the first layer, $\sigma(.)$ is the activation function[2] of the second (output) layer, $k = 1, ..., K$ is the total number of outputs, $b_2 = w_{k0}^{(2)}$ is the bias parameter of the hidden layer. We absorbed the bias parameters into the set of weight parameters, so that Eqn.. 4.3f represents the final form of a 2-layer ANN. Thus, the neural network model is a non-linear function from a set of input variables $\{x_i\}$, represented by vector $\mathbf{x}$, to a set of output variables y_k, controlled by a vector $\mathbf{w}$ of adjustable parameters.

4.3 Evaluation of the error-function derivative

The evaluation of the error-function derivative quantifies the performance of a predictive model (Brownlee, 2020f). This involves splitting a full dataset into a training, validation and testing set (see 4.5.1). The model is then trained on a training dataset, validated on the validation set for hyperparameter tuning, and then tested on a holdout testing set. The predicted values are then compared to the target values (Ferri et al., 2009; Sun et al., 2009).

We evaluate our models using a variety of classification evaluation metrics[3,4,5]. These include (in alphabetical order): accuracy (4.3.1), confusion matrix (4.3.2), F1-score (4.3.3), false negative rate (FNR) (4.3.4), log loss (4.3.5), mean absolute error (MAE) (4.3.6), precision (4.3.7), and recall (4.3.8). Moreover, we calculate three variations of precision, recall, and F1-score, which include: (i) per class label, (ii) macro average, and (iii) weighted average[6].

[1] Jain (2019), URL: https://www.linkedin.com/pulse/activation-functions-neural-networks-rahul-jain.

[2] Kathuria (2018), URL: https://blog.paperspace.com/vanishing-gradients-activation-function/.

[3] Harrel (2019), https://www.fharrell.com/post/classification/.

[4] The taxonomy proposed by Ferri et al. (2009), divides the evaluation metrics into three groups: (i) threshold metrics, (ii) ranking metrics, and (iii) probability metrics. Threshold metrics quantify the classification prediction errors, and include accuracy, confusion matrix, F-score, mean absolute error (MAE), precision and recall. Ranking metrics evaluate classifiers based on how effective they are at separating classes, such as the false negative rate. Probabilistic metrics quantify the uncertainty in predictions, such as log loss (Brownlee, 2020f; Ferri et al., 2009). Brownlee (2020f), URL: https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/.

[5] We evaluate our models with a variety of evaluation metrics to ensure a consistent performance of our proposed deep learning approach for patent valuation. Given the high imbalance nature of the problem, i.e. highly positive skewness of distributions of forward citations, especially when $T = 4$, we evaluate the performance of our model, with transparent reporting, on a variety of metrics to increase our confidence in its predictive ability (Brownlee, 2020c; Ferri et al., 2009; Provost, 2000; Sun et al., 2009). Previous studies only focus on a selective number of metrics (see 2.2.2.3.2 and Table 2.12). Brownlee (2020c), URL: https://machinelearningmastery.com/failure-of-accuracy-for-imbalanced-class-distributions/.

[6] The macro average calculates metrics for each label, and finds their unweighted mean. This does not take into account class imbalance. The weighted average calculates metrics for each label, and finds their

4.3.1 Accuracy

Accuracy is the most widely used metric to evaluate a classifier (Lee et al., 2018). It is defined as the degree of right predictions of a model, i.e. the number of correct predictions over the total predictions (Ferri et al., 2009). We calculate the average accuracy with Eqn 4.4 (see 4.3.2 for definition of terms):

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{4.4}$$

4.3.2 Confusion matrix

The confusion matrix (Table 4.2) is a summary of prediction results on a classification task (Sokolova & Lapalme, 2009). For our models a true positive (TP) is when the patent is low value, V_L and is predicted by the model to be low value, V_L. A true negative (TN) is when the patent is high value, V_H and is predicted by the model to be high value, V_H.

Table 4.2 Confusion matrix

Confusion Matrix		Target variable	
		High value, V_H	Low value, V_L
Predicted variable	High value, V_H	True negative (TN)	False negative (FN)
	Low Value, V_L	False positive (FP)	True positive (TP)

In addition, a false positive (FP) is when a patent is high value, V_H but is predicted to be low value, V_L. Thus, one can interpret that a false positive (FP) represents a missed opportunity, i.e. when a patent is high value, V_H but it remains unexploited because the model predicts it to be low value, V_L, and the firm's management decides against exploiting it (Baglieri & Cesaroni, 2013; Gregory, 1995).

Moreover, a false negative (FN) is when a patent is low value, V_L but is predicted to be high value, V_H. Thus, one can interpret that a false negative represents a wrong investment, i.e. when a patent is low value V_L but it is heavily exploited, with resource commitment and development investment because the model predicts it to be high value, V_H and the firm's management decides on fully exploiting it (Arora et al., 2008; Ernst, 1995; Soenksen & Yazdi, 2016; Verbano & Nosella, 2010). This has more serious implications for firms, since committing resources, and investing for the development of patented inventions to tangible outputs might lead to depletion of resources or financial losses (Benaroch, 2001; Bond & Meghir, 1994; Gambardella et al., 2011; Greenhalgh & Rogers, 2006; Griliches, 1998).

weighted average by taking into consideration the number of datapoints in each class. This alters the macro average to account for class imbalance (Pedregosa et al., 2020; Shmueli, 2019). Pedregosa et al. (2020), URL: https://scikit-learn.org/stable/modules/model_evaluation.html.

4.3.3 F1-score

The F1-score[1] is the harmonic mean of precision and recall. It gives equal weight to both precision and recall (Ferri et al., 2009). We calculate the F1-score with Eqn 4.5, using the definitions of precision (4.3.7) and recall (4.3.8):

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4.5}$$

4.3.4 False negative rate (FNR)

The false negative rate (FNR) is the number of false negatives[2] over the total number of positives (TP + FN). We calculate the FNR with Eqn 4.6 (see 4.3.2 for definition of terms):

$$False\ negative\ rate,\ (FNR) = \frac{FN}{TP + FN} \tag{4.6}$$

4.3.5 Log loss

The most common metric for evaluating predicted probabilities is log loss for classification[3]. It measures how good probability estimates are and is used when calibration and stability is important (Ferri et al., 2009). We calculate the log loss for every iteration cycle (epoch) of training and validation and compare the development of the learning curves[4] (see 4.4.2).

4.3.6 Mean absolute error (MAE)

Mean absolute error (MAE) measures the average magnitude of how much a set of predictions deviates from the target true values.

4.3.7 Precision

Precision is the ability of a classification model to return only relevant instances[5], i.e. the number of correctly classified positives (TP) over the total number relevant instances (TP +

[1] Dercksen (2018), URL:https://koendercksen.com/micro-averaged-f1-optimization-using-neural-networks.html.

[2] Valchanov (2018), https://towardsdatascience.com/false-positive-and-false-negative-b29df2c60aca.

[3] Brownlee (2020f), URL: https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/.

[4] Brownlee (2019h), URL: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/.

[5] Saxena (2018), URL: https://towardsdatascience.com/precision-vs-recall-386cf9f89488.

FP). We calculate precision with Eqn 4.7 (see 4.3.2 for definition of terms):

$$Precision = \frac{TP}{TP+FP} \tag{4.7}$$

4.3.8 Recall

Recall, is the ability of a model to find all the relevant cases[1], i.e. the number of correctly classified positives (TP) over the total number of positives (TP + FN). We calculate recall with Eqn 4.8 (see 4.3.2 for definition of terms):

$$Recall = \frac{TP}{TP+FN} \tag{4.8}$$

4.3.9 Classification threshold (Θ)

Classification predictive modelling involves predicting a class label. This is achieved by using a threshold (Θ), which is also known as classification threshold or decision threshold[2], where the model converts the probability returned into a class (Lipton et al., 2014; Provost, 2000). The default threshold (Θ) is 0.50, where all values greater than the threshold are mapped to one class and all other values are mapped to another class. For classification problems with high imbalanced datasets (Haibo He & Garcia, 2009), we tune the threshold (Θ) as part of a post-processing step approach, where we convert the outputs of a classifier into optimal predictions (Lipton et al., 2014). We tune the classification threshold (Θ) to optimise the prediction outputs by maximising the macro average F1-score (Ferri et al., 2019; Lipton et al., 2014; Zou et al., 2016).

4.4 Network optimisation

Deep learning has recently enjoyed success across a variety of complex problems, for example the synthesis prediction for drug discovery (Chen et al., 2018). Artificial neural networks (ANN) are able to recover good solutions that minimize the error, controlled by a number of adjustable parameters. These models are composed of two different types of parameters: (i) the hyperparameters, which are all the arbitrarily defined parameters, and (ii) the model parameters, which are learned during the model training, i.e. the weights that define how to

[1] Koehrsen (2018b), URL: https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c.

[2] Brownlee (2020b), URL:https://machinelearningmastery.com/threshold-moving-for-imbalanced-classification/.

use the input data to get the desired output. Hyperparameters, through fine tuning[1], determine the structure of the model to identify the network's optimised form for generalisation using the model parameters (Jääsaari et al., 2019; Neyshabur et al., 2017).

The problem of tuning these hyperparameters is solved by autotuning algorithms known as search methods[2,3]. These search methods, such as grid search[4], random search[5], or trial and error search, are autotuning algorithms,which find the optimal hyperparameter values to optimise an evaluation metric (see 4.3) (Bergstra et al., 2011, 2013; Komer et al., 2014).

In our study, we take a quasi-experimental approach, using all 3 search methods for tuning and optimising our networks for the chosen output feature proxies (see Table 4.1). Fig. 4.3 shows the top level quasi-experimental approach we follow. Firstly, we use a combination of the grid search and random search autotuning algorithms, to determine an initial set of evaluated hyperparameters[6]. Secondly, we fine tune the performance of our model using a combination of random search and trial and error (Bergstra et al., 2011; Brownlee, 2019k). We do this by using the learning curve approach[7,8] to diagnose the performance of the model, and identify performance problems (Domhan et al., 2015).

[1]Brownlee (2019l), URL: https://machinelearningmastery.com/hyperparameters-for-classification-machine-learning-algorithms/,

[2]Ippolito (2019), URL: https://towardsdatascience.com/hyperparameters-optimization-526348bb8e2d.

[3]Koehrsen (2018a), URL: https://towardsdatascience.com/automated-machine-learning-hyperparameter-tuning-in-python-dfda59b72f8a.

[4]In grid search, all the combinations of hyperparamters are implemented, and the model is evaluated. The pattern follows a grid matrix, where the model with the highest accuracy is considered the best (Senapati, 2018). One of the major drawbacks of grid search is that it suffers from the dimensionality problem, i.e. when the number of hyperparameters increases, the number of models to be evaluated increases exponentially (Brownlee, 2016b). Brownlee (2016b), URL: https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models-python-keras/.

[5]In random search, random combinations of a range of the hyperparameters are used to find the best solution for the built model. The probability of finding the optimal parameters are comparatively higher in random search because of the random search pattern where the model might end up being trained on the optimised parameters without any aliasing (Senapati, 2018)

[6]The initial range of hyperparameters is based on previous literature in Tables 2.10 and 2.12 to determine the grid's start. We then train the quasi-search models for 30 epochs (iterations) because of the time constrains. However, if we train the quasi-search models to convergence, i.e. a large number of training epochs, then early stopping occurs around 30 epochs (Brownlee, 2019j; Caruana et al., 2001; Prechelt, 2012). Early stopping is the point at which the performance of the classifier evaluated against the validation set, begins to degrade, i.e. log loss begins to increase or accuracy begins to decrease, at which point the training process is stopped (Bishop, 2006).

[7]Brownlee (2019h), URL: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/.

[8]A learning curve is a plot of model learning performance over experience or time. Learning curves are widely used in performance diagnosis. The evaluation of each model on the training dataset and on a holdout validation dataset after each update (or epoch) during training is then plotted. Reviewing the learning curves and the model performance during training can be used to diagnose problems with learning, such as underfitting or overfitting, as well as whether the training and validation datasets are suitably representative (Brownlee, 2019h; Domhan et al., 2015).

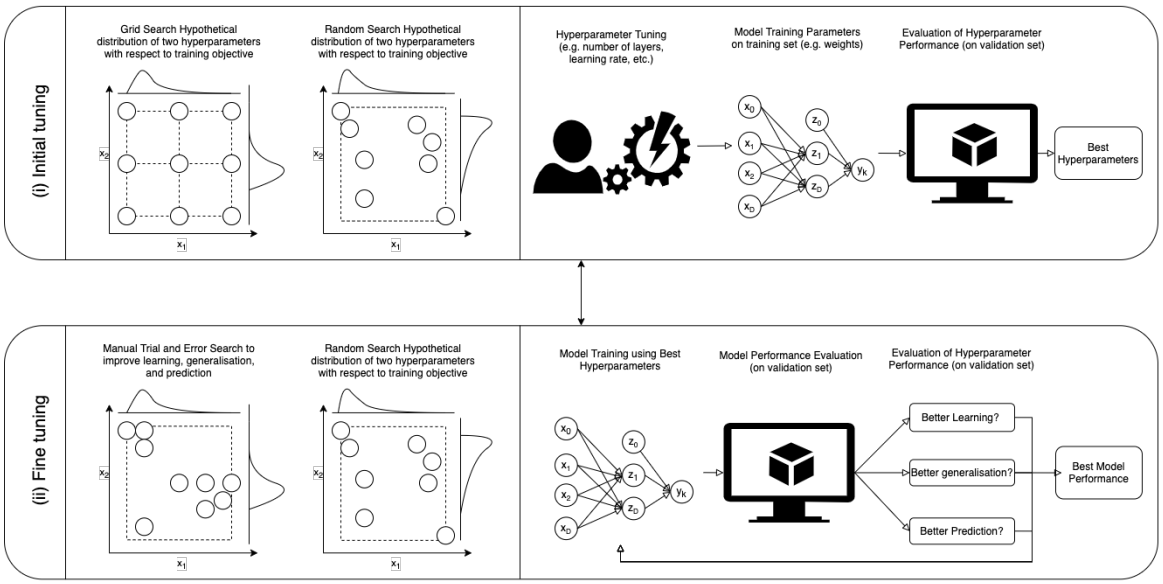


Fig. 4.3 Quasi-experimental approach for network optimisation, using grid search, random search, and trial and error

There are 3 types of problems that are straightforward to diagnose for a poorly performing deep learning model (Brownlee, 2019b). These include: (i) problems with learning, where a model cannot effectively learn from the training dataset or shows slow progress; (ii) problems with generalisation, where a model overfits the training dataset and performs poorly on a holdout test dataset, and (iii) problems with prediction, where the training algorithm has a strong influence on the final model, causing a high variance in behaviour and performance. We follow a 3-step framework approach, proposed by Brownlee (2019e), for identifying and improving problems with performance: (i) better learning, with methods that improve the adaptation of neural network model weights in response to a training dataset; (ii) better generalization, with methods that improve the performance of a neural network model on a test dataset; (iii) better predictions, with methods that reduce the variance in the performance of a final model[1,2].

In the first step, we identify the initial set of hyperparameters using a combination of grid search and random search methods. Fig. 4.4 shows the results of the initial tuning of the quasi-experimental approach in Fig. 4.3. The colours represent the number of different experiments[3]. The plots in Fig. 4.4 include: 4.4a validation loss vs. training loss; 4.4b

[1]Brownlee (2019e), URL: https://machinelearningmastery.com/framework-for-better-deep-learning/.

[2]Maheswari (2019), URL: https://towardsdatascience.com/breaking-the-curse-of-small-data-sets-in-machine-learning-part-2-894aa45277f4.

[3]Due to the large number of experiments produced from grid search and random search autotuning experiments, with trying combinations of hyperparameters, a key is omitted from Fig. 4.4. This is because, for

validation accuracy vs. training accuracy; 4.4c training precision vs. training recall; 4.4d validation precision vs. validation recall; 4.4e validation F1-score vs. training F1-score.

From Fig. 4.4a, we observe that there is a non-linear relationship between training loss and validation loss due to the evaluation of different hyperparameters. The training loss should be about the same as the validation loss, and a near-to linear relationship should exist, indicating that learning takes place[1]. Fig. 4.4b shows the training accuracy vs. validation accuracy, which has a logarithmic trend relationship, indicating that either the model is overfitting or the validation dataset is not representative of the training dataset, i.e. the distributions of the training and validation datasets are different[2]. However, we also observe a set of experimental values above the trend line, indicating that the model is learning from the training dataset with the experimental hyperparameter combinations. This is probably due to the increase on the network size and the introduction of regularisation (Kukačka et al., 2017; Ng, 2004). These experiments also coincide with the reduction in the loss function (Fig. 4.4a), and improvements in identifying more easily the elements of the confusion matrix (Fig. 4.4c and 4.4d). We also observe that the trend curve is higher in Fig. 4.4d than Fig. 4.4c, attributed to the introduction of the dropout regularisation method (see 4.4.1.2). This is also supported by Fig. 4.4e with a few experiments above the trend line indicating the increase in the model's learning ability.

example for two hyperparameters, each testing 3 values, there are 9 combinations of experiments, as shown in Fig. 4.3. Due to the large number of hyperparameter testing, which produced a large number of experiments, we omit the key to avoid confusion. We plot all the experiments on the same set of axis, to identify patterns in the model's performance and hyperparameters before we fine tune the models (see Fig. 4.3). Some experiments are overlayed because of the similarity in the results.

[1]Brownlee (2019h), URL: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/.

[2]The shape and dynamics of a learning curve can be used to diagnose the behaviour of a model and in turn identify configuration changes to improve learning and performance. There are three dynamics to be observed: underfit, overfit, good fit. Underfitting refers to a model that cannot learn from the training dataset, and occurs when the model is not able to obtain a sufficiently low error value on the training dataset (Goodfellow et al., 2016). An underfitted model can be identified either from a flat learning curve of the training loss or from the training accuracy curve being significantly lower than the validation accuracy curve (Brownlee, 2019h). Overfitting occurs when a models learns the training dataset 'too well', i.e. memorises it (Domhan et al., 2015). The more overfitted the model is, the less it is able to generalise to new data. This increases the generalisation error and occurs either when the model has more capacity, i.e. flexibility, than is required for the problem or if prolong training occurs (Murphy, 2012). A good fit is the target of the learning algorithm and exists between an overfit and underfit model. A good fit is identified by a training and validation loss function that decreases to a point of stability with a minimal gap between the two final loss values. The model's training loss will always be slightly lower than the validation dataset, which is known as generalisation gap (James et al., 2013).

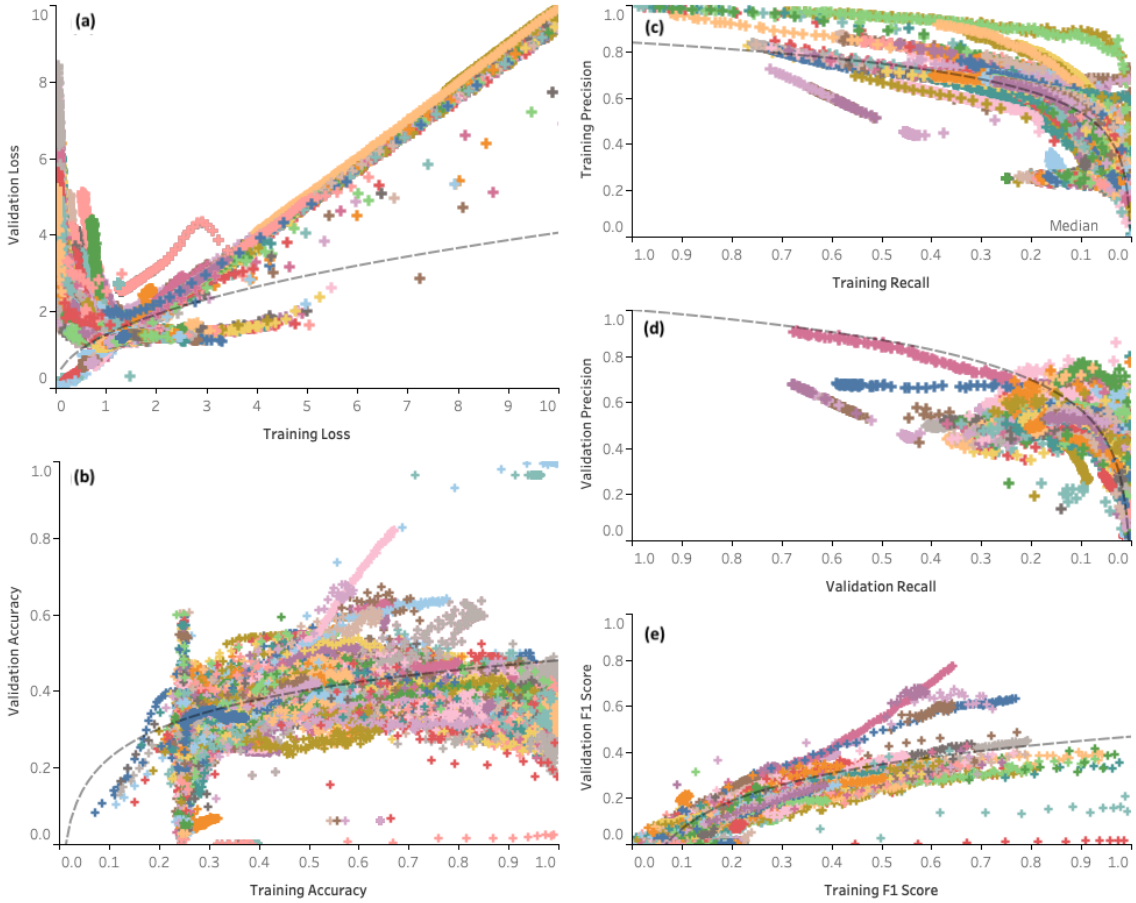


Fig. 4.4 Plots of results from the initial tuning (grid search and random search, see Fig. 4.3): 4.4a validation loss vs. training loss; 4.4b validation accuracy vs. training accuracy; 4.4c training precision vs. training recall; 4.4d validation precision vs. validation recall; 4.4e validation F1 score vs training F1 score. The colours represent the different number of experiments

In the second step, we fine tune the network using a combination of random search methods and trial and error methods approach, to determine optimal networks (see Fig. 4.3). Fig. 4.5 shows the results of all the experiments for the fine tuning of the network. These are aggregated results of all experiments, of which a selection is reported in 4.4.1 and 4.4.2 to show the development of the optimisation of the deep neural network's parameters[1].

[1]In Fig. 4.5, we plot all the experiments on the same set of axis, to identify patterns in the model's performance and identify the best model (see Fig. 4.3). The colours represent some of the experiments. We omit the key to avoid confusion because of the number of experiments, which is smaller than the ones in Fig. 4.4 since we have already identified a set of initial hyperparameters from step 1 (see Fig. 4.3). Some experiments are overlayed because of the similarity in the results. A selection of experiments to show the development of the optimisation of the deep neural network's parameters to identify the best model are reported in 4.4.1.

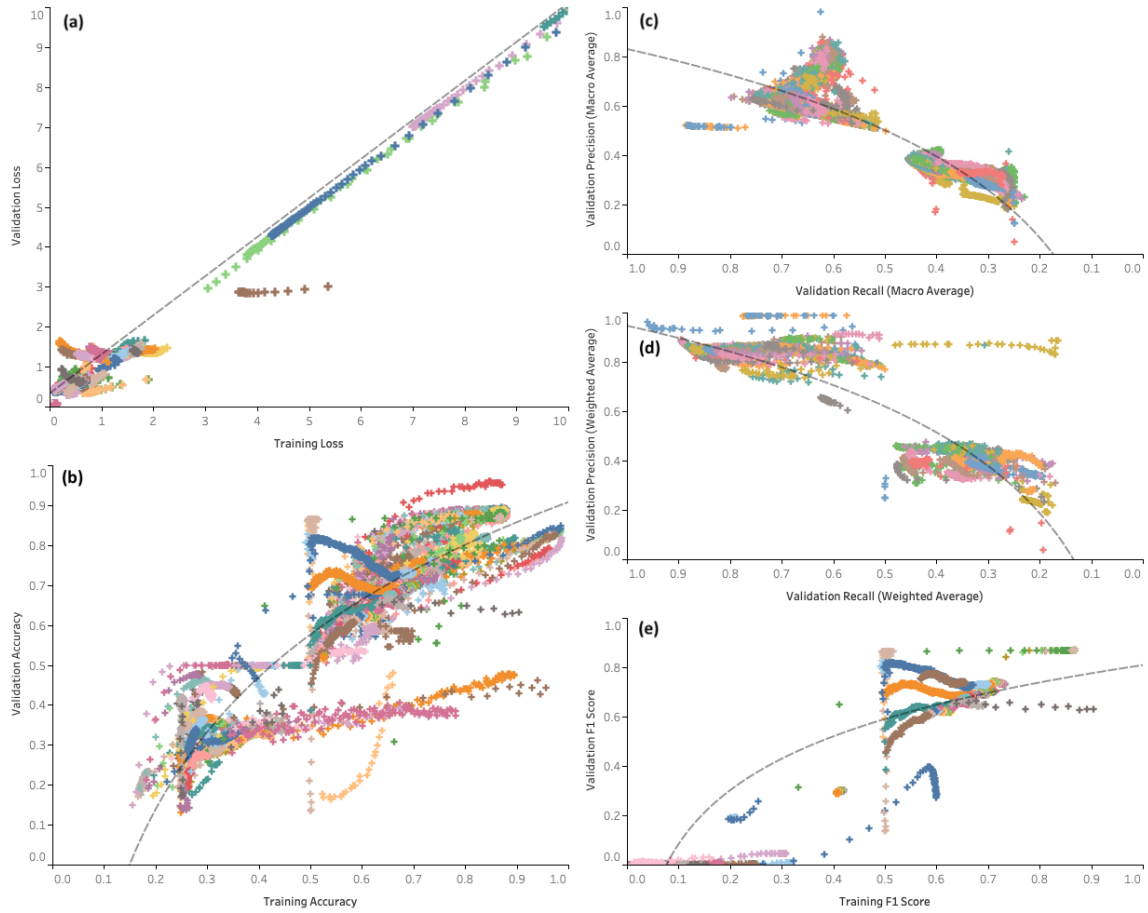


Fig. 4.5 Plots of results from the fine tuning (trial and error and random search, see Fig. 4.3): 4.5a validation loss vs. training loss; 4.5b validation accuracy vs. training accuracy; 4.5c validation precision (macro average) vs. validation recall (macro average); 4.5d validation precision (weighted average) vs. validation recall (weighted average); 4.5e validation F1-score vs training F1-score. The colours represent the different number of experiments

From Fig. 4.5a validation loss vs. training loss, we observe that the results of these experiments are concentrated along the trend line, which indicates that the ANN is learning, with sometimes minimal underfitting (some of the experiments are further up the trend line). This is also supported by Fig. 4.5b validation accuracy vs. training accuracy, where the accuracies increase of the quasi-models, with a high concentration of experiments concentrated above the trend line in the upper right quadrant. The improved learning and generalising ability of the models can be observed by Fig. 4.5c, Fig. 4.5d, and Fig. 4.5e, where the evaluation metric of precision, recall and F1-score have been increasing. These are also above the trend lines, and concentrated on the upper left quadrants.

4.4.1 Parameter optimisation

We optimise the hyperparameters of the artificial neural networks, which have an impact on the model's parameters. Hyperparameters determine the structure of the model to identify the network's optimised form for generalisation using the model parameters, the weights (Neyshabur et al., 2017). Firstly, we focus on the neural network capacity (4.4.1.1), to determine the number of layers and nodes for our model. Then, we reduce the generalisation error by regularisation methods, such dropout, batch normalisation, and L2 penalty (4.4.1.2), to improve and tailor our model's ability to learn and generalise to new data.

4.4.1.1 Neural network capacity

The capacity[1] of a deep learning ANN controls the scope of the types of mapping functions that it is able to learn[2].Increasing the number of layers provides an increase to the capacity of the model[3]. Neural networks, with high capacity are known as deep neural networks[4]. Deep neural networks with a high number of layer nodes[5] are known as wide and deep neural networks (Pandey & Dukkipati, 2014). We need to optimise the capacity of our model, so it fits with our dataset's size and complexity (see 3.3), with the aim to learn how to map the inputs to the outputs. In the experiments described below, we focus on the number of layers, and number of nodes in the hidden layers. By adding more layers and nodes within a layer, a deep network can represent functions of increasing complexity (Goodfellow et al., 2016).

From Tables 2.10 and 2.12, current research focuses on shallow 2 and 3 layer neural networks[6,7], with only a limited number of recent studies using a slightly deeper network structures (see 2.2.2.3.2). Thus, we use ANNs with more than 4 layers. In addition, we

[1]The capacity of a neural network is defined as configuration of neurons or nodes and layers, i.e. the number of layers, the number of input nodes, the number of output nodes, and the number of nodes in each layer (Brownlee, 2019g; Hopfield, 1982; Jia et al., 2016). A model with too little capacity cannot learn the training dataset, and underfits, whereas a model with too much capacity memorises the training dataset and thus overfits.

[2]Brownlee (2019g), URL: https://machinelearningmastery.com/how-to-control-neural-network-model-capacity-with-nodes-and-layers/.

[3]A model with more nodes or more layers has a greater capacity and is potentially capable of learning a larger set of mapping functions, governed by the data complexity.

[4]Deep neural networks are defined as networks with architectures with multiple hidden layers, i.e. high capacity, where as shallow neural networks have one or two hidden layer (Delalleau & Bengio, 2011; Goodfellow et al., 2016; Murphy, 2012).

[5]The number of layer nodes is referred to as the width, and the number of layers is referred to as the depth, of the neural network (Abood & Feltenberger, 2018; Brownlee, 2019g).

[6]We adopt the terminology by Murphy (2012), where a 2 layer artificial neural network (ANN) is a network with 2 layers of adaptive weights, i.e. 1 hidden layer

[7]Shallow neural networks have 1 or 2 hidden layer, i.e they are a 2 or 3 layer network respectively (Delalleau & Bengio, 2011; Goodfellow et al., 2016; Murphy, 2012).

set the number of nodes in the input layer to the number of input features. There is no universally accepted method for setting the number of nodes in the hidden layers[1], with only some heuristic methods having been reported in the literature (Kavzoglu & Mather, 2003; Stathakis, 2009). We follow a systematic experimentation approach[2], where we combine the following two approaches: (i) the heuristics, proposed by Heaton (2005), where the number of hidden nodes should be between the size of the input layer and the size of the output layer, should be 2/3 the size of the input layer, plus the size of the output layer, and should be less than twice the size of the input layer; (ii) the argument proposed by Goodfellow et al. (2016) to go for a greater depth and width network for a prediction model, where the data size and complexity is high (Chen et al., 2019; Hornik, 1991; Zhang et al., 2016a). Thus, we set the number of nodes in the hidden layers to 2048 nodes, and allow the model to try numbers in the range 128-4096 and determine the number of nodes, based on the best performance.

Fig. 4.6[3] shows the network capacity tuning experiments. This is complemented by Table 4.3. From Fig. 4.6, we observe that the training and validation loss curves follow each other, which shows that the deep ANN's learning is stable (Brownlee, 2019h). This is supported by Table 4.3, since with increasing capacity the training loss and validation loss are similar. The training accuracy is slightly lower that the validation accuracy in Fig. 4.6 and in Table 4.3.

From Table 4.3, as the number of layers increases, the validation accuracy increases. This saturates at 7 layers, since any more layer addition causes the network to overfit (training accuracy rises above the validation accuracy), while validation precision (macro average) and validation F1-score (macro average) drop significantly (see Table 4.3 models h and i). We can also observe that increasing the width of the network (no. of nodes), allows the ANN to capture the complex relationships of the dataset, since the training loss and validation loss drop even further (see Table 4.3 models e, f and g). Also, the weighted averages for precision, recall, and F1-score are higher than the macro averages for precision, recall, F1-score, because they take into consideration the class imbalance present within the data. Table 4.3 model g from is slightly underfitting since the training accuracy is less than the validation accuracy. However, it has the highest validation F1-score (macro), which indicates that it has more capacity to learn and requires further tuning to reduce the generalisation error. Thus, Table 4.3 model g with 7 layers and 2048 nodes on each layer, is a wide and deep

[1]Cross Validated (2010), URL: https://stats.stackexchange.com/questions/181/how-to-choose-the-number-of-hidden-layers-and-nodes-in-a-feedforward-neural-netw.

[2]Brownlee (2017e), URL: https://machinelearningmastery.com/evaluate-skill-deep-learning-models/.

[3]Fig. 4.6 is a dynamic representation of the experiments, i.e. shows the evaluation of the different configuration models per epoch (iteration cycle) on the training and the validation datasets. It shows a selective number of experiments, which are overlayed due to the similarity in the results, and a sub-selection of those are reported in Table 4.3. Table 4.3 is a static representation of the experiments, i.e. shows the evaluation of the different configuration models at the last epoch (end of training) on the training and validation datasets.

ANN, with stable predictive power, and forms the basis for the neural network regularisation experiments (4.4.1.2).

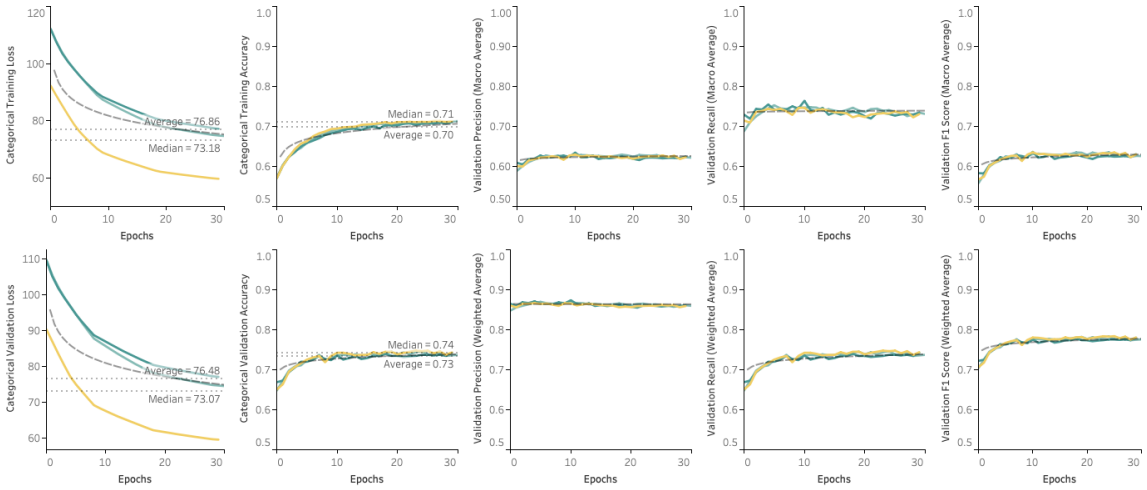


Fig. 4.6 Dynamic (per epoch) network capacity tuning evaluation results for selective experiments: training loss, validation loss, validation precision (macro average), validation recall (macro average), validation F1-score (macro average), validation precision (weighted average), validation precision (weighted average), validation F1-score (weighted average)

Table 4.3 Static end of training (last epoch) network capacity tuning evaluation results for a selection of experiments from Fig. 4.6

Trial[a,b,c]	Layers[d]	Nodes[e]	Training[f] loss	Validation loss	Training accuracy	Validation accuracy	Validation precision (weighted)	Validation recall (weighted)	Validation F1-score (weighted)	Validation precision (macro)	Validation recall (macro)	Validation F1-score (macro)
a	4	2048	73.93	73.91	0.68	0.68	0.88	0.68	0.75	0.56	0.67	0.61
b	5	2048	75.20	75.15	0.68	0.69	0.89	0.69	0.75	0.57	0.69	0.62
c	6	2048	67.99	67.93	0.69	0.70	0.89	0.70	0.76	0.57	0.69	0.63
d	6	4096	52.63	52.63	0.70	0.70	0.88	0.70	0.76	0.57	0.69	0.62
e	7	128	74.79	74.70	0.71	0.74	0.86	0.74	0.77	0.62	0.70	0.65
f	7	1024	77.24	77.15	0.71	0.74	0.86	0.74	0.78	0.62	0.70	0.66
g	7	2048	59.67	59.60	0.71	0.74	0.86	0.74	0.78	0.62	0.73	0.67
h	8	2048	0.38	0.61	0.79	0.76	0.99	0.76	0.85	0.52	0.88	0.65
i	8	2048	66.49	66.55	0.79	0.77	0.99	0.77	0.86	0.52	0.88	0.65

[a] The model with the optimal configuration parameters is shown in bold for this set of experiments.

[b] A selection of experiments from Fig. 4.6 is reported for simplicity.

[c] All experiments have been standardised with the following hyperparamters: no. of input nodes = no. of input features, loss function = categorical cross-entropy, optimiser = Adam, learning rate = $1e^{-7}$, layer activation function = ReLU, output activation function = softmax, batch size = 512, no. of output nodes = 2.

[d] We adopt the terminology by Murphy (2012), where a n layer ANN is a network with n layers of adaptive weights, for the no. of layers.

[e] Heaton (2017), URL: https://www.heatonresearch.com/2017/06/01/hidden-layers.html.

[f] The evaluation metrics are introduced in 4.3 and the dataset terminology of training, validation and testing is introduced in 4.5.1.

4.4.1.2 Neural network regularisation

Training a deep neural network that can generalise[1] well on new data is a challenging problem (Goodfellow et al., 2016). A model with too little capacity cannot learn the problem, whereas a model with too much capacity can learn it too well and overfit the training dataset[2]. A modern approach to reducing the generalization error is to use a deeper model that may require regularization during training to keep the weights of the model small. Methods that seek to reduce the generalization error by keeping network weights small are referred to as regularisation methods, constraining the network complexity and leading to faster optimisation and improved performance (Bishop, 2006; Kukačka et al., 2017; Zhang et al., 2016a). We use 3 methods that are considered regularisation methods: (i) dropout[3] (4.4.1.2.1), (ii) L2 regularisation[4] (4.4.1.2.2), and (iii) batch normalisation[5] (4.4.1.2.3).

4.4.1.2.1 Dropout regularisation

Deep ANN with large number of parameters are very powerful machine learning systems. Dropout[6] randomly drops nodes with their connections from the ANN during training to avoid overfitting (Labach et al., 2019), and prevents them from over-adapting (Gal & Ghahramani, 2015, 2016; King & Zeng, 2003). Dropout forces ANNs to learn useful robust features in relation to random subsets of neighbouring neurons (Ranjan, 2019; Srivastava et al., 2014).

Fig. 4.7 shows the network regularisation tuning experiments for dropout, which is complemented by Table 4.4[7]. We observe that the training and validation loss curves follow each other, indicating a stability in the model's learning ability (Fig. 4.7), with no overfitting and minimal underfitting[8].

[1]The objective of ANN is to have a model that performs well both on the training data and the new data on which the model will be used to make predictions. This is also known as model generalisation or generalising ability (Goodfellow et al., 2016). A model with a near-infinite number of examples will eventually plateau in terms of what the capacity of the network is capable of learning.

[2]Brownlee (2018d), URL: https://machinelearningmastery.com/introduction-to-regularization-to-reduce-overfitting-and-improve-generalization-error/.

[3]Brownlee (2018a), URL: https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/.

[4]Jane Street (2019), URL: https://blog.janestreet.com/l2-regularization-and-batch-norm/.

[5]Brownlee (2019c), URL: https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/.

[6]Maklin (2019), URL: https://towardsdatascience.com/machine-learning-part-20-dropout-keras-layers-explained-8c9f6dc4c9ab

[7]Fig. 4.7 is a dynamic representation of the experiments, i.e. shows the evaluation of the different configuration models per epoch (iteration cycle) on the training and the validation datasets. It shows a selective number of experiments, which are overlayed due to the similarity in the results, and a sub-selection of those are reported in Table 4.4. Table 4.4 is a static representation of the experiments, i.e. shows the evaluation of the different configuration models at the last epoch (end of training) on the training and validation datasets.

[8]Budhiraja (2016), URL: https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-

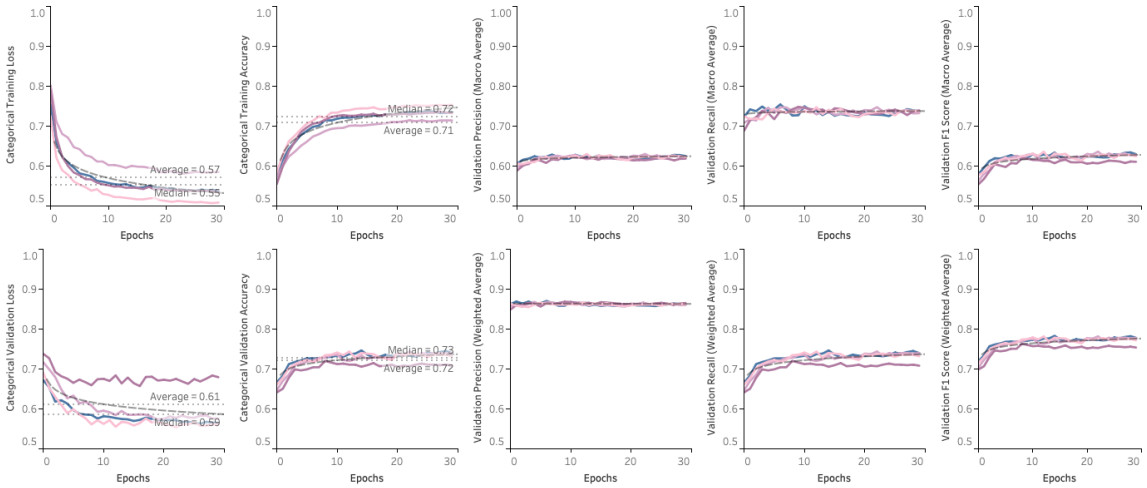


Fig. 4.7 Dynamic (per epoch) network regularisation tuning evaluation results with dropout for selective experiments: training loss, validation loss, validation precision (macro average), validation recall (macro average), validation F1-score (macro average), validation precision (weighted average), validation precision (weighted average), validation F1-score (weighted average)

Table 4.4 Static end of training (last epoch) network capacity tuning evaluation results with dropout for a selection of experiments from Fig. 4.7

Trial[a,b,c]	Dropout (Hidden Layers)[d]	Dropout (Output Layer)[e]	Training loss[f]	Validation loss	Training accuracy	Validation accuracy	Validation precision (weighted)	Validation recall (weighted)	Validation F1-score (weighted)	Validation precision (macro)	Validation recall (macro)	Validation F1-score (macro)
a	0.10	0.00	0.53	0.68	0.73	0.71	0.87	0.71	0.75	0.62	0.74	0.67
b	0.20	0.00	0.59	0.61	0.71	0.73	0.86	0.73	0.77	0.62	0.74	0.67
c	0.40	0.10	0.51	0.56	0.75	0.73	0.86	0.73	0.77	0.62	0.74	0.67
d	0.20	0.10	0.59	0.58	0.71	0.74	0.86	0.74	0.78	0.62	0.73	0.67
e	0.20	0.40	0.68	0.62	0.69	0.74	0.86	0.74	0.78	0.62	0.74	0.68
f	0.20	0.20	0.54	0.56	0.74	0.74	0.86	0.74	0.78	0.62	0.74	0.68
g	0.00	0.00	0.59	0.55	0.72	0.76	0.86	0.76	0.79	0.60	0.71	0.65

[a] The model with the optimal configuration parameters is shown in bold for this set of experiments.

[b] A selection of experiments from Fig. 4.7 is reported for simplicity.

[c] All experiments have been standardised with the following hyperparamters: no. of input nodes = no. of input features, loss function = categorical cross-entropy, optimiser = Adam, learning rate = $1e^{-7}$, layer activation function = ReLU, output activation function = softmax, batch size = 512, no. of output nodes = 2, no. of layers = 7, no. of nodes in hidden layers = 2048.

[d] Dropout is applied between the hidden layers. Brownlee (2018a), URL: https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/.

[e] Dropout is applied between the last hidden layer and the output layer. Brownlee (2016a), URL: https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/.

[f] The evaluation metrics are introduced in 4.3 and the dataset terminology of training, validation and testing is introduced in 4.5.1.

learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5.

However, the validation loss curve is less smooth because the model becomes more sensitive to the validation dataset's distribution. From Table 4.4, we observe that with a 0.20 dropout rate (i.e. 20% of nodes dropped randomly per layer) in all layers (hidden and output layers), there is optimal performance with the training and validation loss, which are close to each other (Table 4.4 model f). This is supported by the training and validation accuracy, with the introduction of dropout yielding stable results. However, Table 4.4 models e and g exhibit overfitting. Table 4.4 model f becomes less sensitive to the node weights (Srivastava et al., 2014), increasing its capability of better generalisation and less likely to overfit the training data[1], and forms the basis for the ANN L2 penalty regularisation experiments (4.4.1.2.2).

4.4.1.2.2 L2 penalty regularisation

The ANN learns a set of weights **w** (see Eqn. 4.3f) using a gradient descent derivative on the training dataset, during fitting. Large weights make the ANN unstable, causing sharp transitions in the node functions and thus large changes in the output for small changes in the inputs. This indicates an overly specialised ANN to training data (Goodfellow et al., 2016). Having small weights allows the model to focus learning. The learning algorithm (4.4.2) can be updated to encourage the ANN towards small weights by introducing the size of the weights as a penalty, i.e. penalising the model's loss function (4.4.2.2) proportional to the weights' size[2] (Ng, 2004).

We use the L2 penalty, where the sum of squared weights **w** is added into the loss function as a penalty term (λ) to be minimised[3] (Hastie, Trevor, Tibshirani, Robert, Friedman, 2009; Murphy, 2012). The λ hyperparameter[4] controls the amount of bias in the model between 0.0 (no penalty, low bias and high variance) and 1.0 (full penalty, high bias and low variance)[5,6].

[1] Brownlee (2016a), URL: https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/.

[2] Brownlee (2018g), URL: https://machinelearningmastery.com/weight-regularization-to-reduce-overfitting-of-deep-learning-models/.

[3] The L2 penalty is also known as ridge regression in other domains (Hoerl & Kennard, 1970; van Wieringen, 2018). It performs a weight decay or shrinkage because it penalises larger weights to decay towards zero unless supported by the data (Bishop, 2006; Cortes et al., 2009; Loshchilov & Hutter, 2019a).

[4] It is computationally inefficient and expensive to search for the correct value of multiple hyperparameters. It is reasonable to use the same weight decay at all layers to reduce the size of search space (Goodfellow et al., 2016), and a good configuration strategy is usually to start with larger networks and use weight decay.

[5] Brownlee (2018f), URL: https://machinelearningmastery.com/how-to-reduce-overfitting-in-deep-learning-with-weight-regularization/.

[6] The model underestimates the weights and underfits the training data with a strong penalty, where as it overfits the training data with a weak penalty (Lau et al., 2020; Oppermann, 2020).

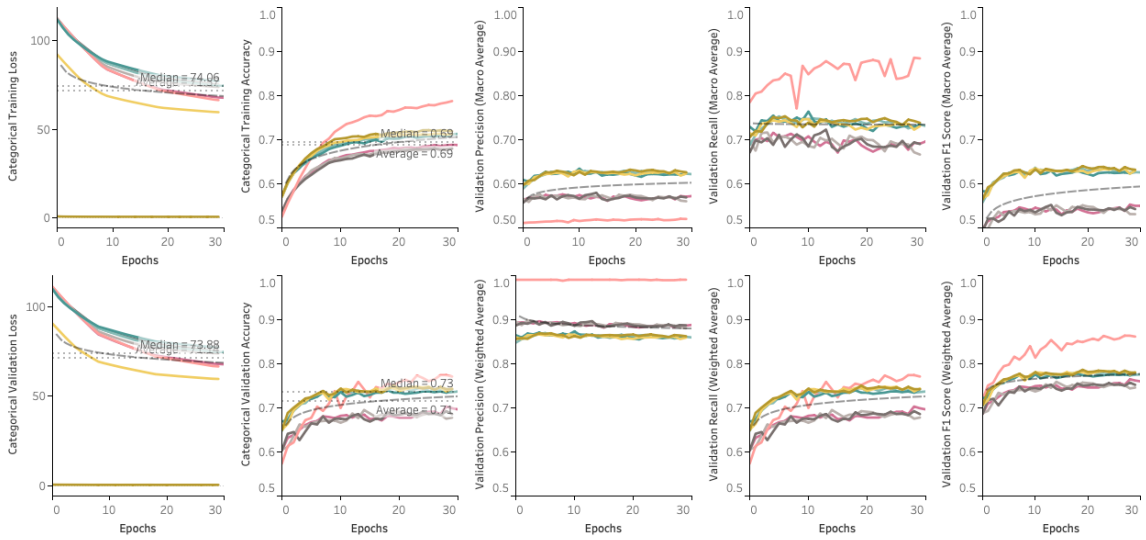


Fig. 4.8 Dynamic (per epoch) network regularisation tuning evaluation results with L2 penalty for selective experiments: training loss, validation loss, validation precision (macro average), validation recall (macro average), validation F1-score (macro average), validation precision (weighted average), validation precision (weighted average), validation F1-score (weighted average)

Table 4.5 Static end of training (last epoch) network capacity tuning evaluation results with L2 penalty for a selection of experiments from Fig. 4.8

Trial[a,b,c]	L2 Penalty[d]	Training[e] loss	Validation loss	Training accuracy	Validation accuracy	Validation precision (weighted)	Validation recall (weighted)	Validation F1-score (weighted)	Validation precision (macro)	Validation recall (macro)	Validation F1-score (macro)
a	0.0001	73.93	73.91	0.68	0.68	0.88	0.68	0.75	0.56	0.67	0.61
b	0.0010	75.20	75.15	0.68	0.69	0.89	0.69	0.75	0.57	0.69	0.62
c	0.0050	67.99	67.93	0.69	0.70	0.89	0.70	0.76	0.57	0.69	0.63
d	0.0070	74.79	74.70	0.71	0.74	0.86	0.74	0.77	0.62	0.73	0.67
f	0.0090	77.24	77.15	0.71	0.74	0.86	0.74	0.78	0.62	0.73	0.67
g	0.0100	59.67	59.60	0.73	0.74	0.86	0.74	0.78	0.62	0.73	0.67
h	0.0200	66.49	66.55	0.79	0.77	0.99	0.77	0.86	0.52	0.88	0.66

[a] The model with the optimal configuration parameters is shown in bold for this set of experiments.

[b] A selection of experiments from Fig. 4.8 is reported for simplicity.

[c] All experiments have been standardised with the following hyperparamters: no. of input nodes = no. of input features, loss function = categorical cross-entropy, optimiser = Adam, learning rate = $1e^{-7}$, layer activation function = ReLU, output activation function = softmax, batch size = 512, dropout rate = 0.2, no. of output nodes = 2, no. of layers = 7, no. of nodes in hidden layers = 2048, dropout rate = 0.2.

[d] We follow the approach by Brownlee (2018g) to introduce the same L2 penalty to all layers. Brownlee (2018f), URL: https://machinelearningmastery.com/how-to-reduce-overfitting-in-deep-learning-with-weight-regularization/.

[e] The evaluation metrics are introduced in 4.3 and the dataset terminology in 4.5.1.

Fig. 4.8[1] shows the network regularisation tuning experiments for L2 penalty, and is complemented by Table 4.5. While the curves are shifted upwards due to the addition of the penalty term λ, we observe a stable training and learning from the smoothness of the training loss curve and the validation loss curve in Fig. 4.8, originating from the weight decay constrain (Hoerl & Kennard, 1970). From Table 4.5, with a value of 0.01 for the L2 penalty, the model's fit seems just right, with a tendency towards undefitting (model c, d and f), and only overfitting (model h) when the value of the L2 penalty is high (Ng, 2004). We also observe an increase in the accuracy curves. Due to the data complexity, we allow the model some flexibility to avoid overfitting (Bishop, 2006; Lau et al., 2020). Thus, we use Table 4.5 model g that exhibits the optimal performance for the next set of experiments with batch normalisation (4.4.1.2.3).

4.4.1.2.3 Batch normalisation regularisation

Training deep ANN is complicated because the distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This is known as internal covariate shift and Ioffe & Szegedy (2015) propose the batch normalisation layer to address it[2], which normalises the layer inputs, and allows the usage of higher learning rates, acting as a form of regularisation[3]. Recently, Chen et al. (2019) expand the method of batch normalisation by introducing the independent component layer, combining batch normalisation and dropout before each weight connection to achieve faster convergence and improved performance (Mhaskar et al., 2017).

We use the batch normalisation layer, together with the dropout layer (4.4.1.2.1) (Srivastava et al., 2014). Deep ANN with batch normalisation are more stable and converge faster[4]. We evaluate the introduction of the batch normalisation layer in relation to the sequence of layers and the activation function (Ioffe & Szegedy, 2015). The activation function[5]

[1]Fig. 4.8 is a dynamic representation of the experiments, i.e. shows the evaluation of the different configuration models per epoch (iteration cycle) on the training and the validation datasets. It shows a selective number of experiments, which are overlayed due to the similarity in the results, and a sub-selection of those are reported in Table 4.5. Table 4.5 is a static representation of the experiments, i.e. shows the evaluation of the different configuration models at the last epoch (end of training) on the training and validation datasets.

[2]Brownlee (2019c), URL: https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/.

[3]Brownlee (2019f), URL: https://machinelearningmastery.com/how-to-accelerate-learning-of-deep-neural-networks-with-batch-normalization/.

[4]The effectiveness of the batch normalisation layer stems from controlling the change of the layers' input distributions during training, making the optimisation landscape significantly smoother and inducing a more predictive and stable behaviour of the gradients (Bjorck et al., 2018; Santurkar et al., 2018).

[5]The activation function (also known as transfer function) is a mathematical gate in between the input feeding the current neuron (node) and its output, going to the next layer. It can be seen as a transformation that maps the input signals into output signals that are needed for the neural network to function, adding a non-linear property to the functions (see Fig. 4.2a). ML Glossary (2020), URL: https://ml-cheatsheet.readthedocs.io/en/

transforms the summed weighted input from the node into the activation of the node or output for that input[1] (LeCun et al., 2012). There are many types of activation functions, with non-linear activation functions being preferred because they allow the nodes to learn more complex structures in the data[2]. We test the 3 common activation functions for hidden layers[3]: the rectified linear units (ReLU), the sigmoid, and the tanh activation functions[4].

Fig. 4.9[5] shows the network regularisation tuning experiments using batch normalisation and the activation functions, and is complemented by Table 4.6. From Fig. 4.9, we observe the ANN's improved learning from the fast convergence and the smoothness of the training and validation loss curves (Zhang et al., 2016a). This is supported by the shift in the training and validation accuracy curves, indicating an increase in the network's generalising ability (Bjorck et al., 2018). The weighted and macro average validation precision, recall and F1-score curves are sensitive to the validation data, which indicates that the choice of the activation function together with the order of layers has a stability effect on the ANN[6] (Zhang et al., 2016a).

From Table 4.6, we observe models a-h are overfitted, where as model i is underfitted, partly because of the order of the layers. The order of layers suggested by Chen et al. (2019), i.e. dense layer - activation layer - batch normalisation layer - dropout layer, and using a deep ANN with hidden layer sigmoid activation functions[7], exhibits the most stable behaviour (Mhaskar et al., 2017). This is represented by model i, which seems to be fitted just about right, with the training loss and validation loss very similar, the training accuracy and validation accuracies close to each other, and the weighted and macro average of the F1-score the highest of all experiments (Santurkar et al., 2018). We use model i from Table 4.6 for the next set of experiments with the error backpropagation optimisation (4.4.2).

latest/activation_functions.html.

[1]DeepAI (2020), URL: https://deepai.org/machine-learning-glossary-and-terms/activation-function.

[2]Brownlee (2019d), URL: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/.

[3]Sharma (2017a), URL: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

[4]For all models, we use the hierarchical softmax activation function in the output layer for classification (Lj Miranda, 2017; Oliinyk, 2017; Roelants, 2020; Samala, 2017).

[5]Fig. 4.9 is a dynamic representation of the experiments, i.e. shows the evaluation of the different configuration models per epoch (iteration cycle) on the training and the validation datasets. It shows a selective number of experiments, which are overlayed due to the similarity in the results, and a sub-selection of those are reported in Table 4.6. Table 4.6 is a static representation of the experiments, i.e. shows the evaluation of the different configuration models at the last epoch (end of training) on the training and validation datasets.

[6]Sharma (2017a), URL: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

[7]Kathuria (2018), URL: https://blog.paperspace.com/vanishing-gradients-activation-function/.

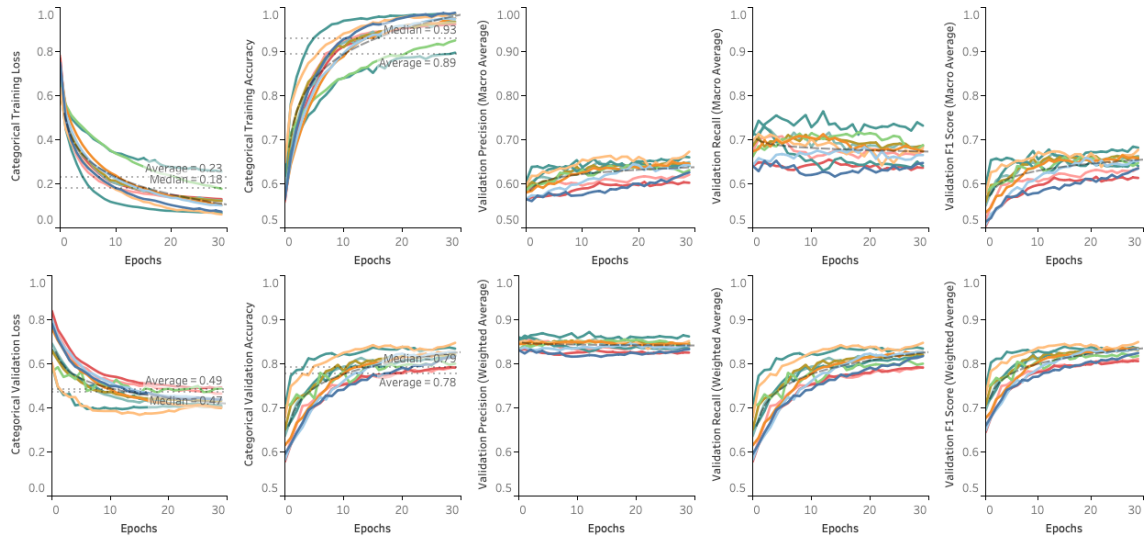


Fig. 4.9 Dynamic (per epoch) network regularisation tuning evaluation results with batch normalisation, order of layers and activation function in dense layers, for selective experiments: training loss, validation loss, validation precision (macro average), validation recall (macro average), validation F1-score (macro average), validation precision (weighted average), validation precision (weighted average), validation F1-score (weighted average)

Table 4.6 Static end of training (last epoch) network regularisation tuning experiments with batch normalisation, order of layers and activation function in dense layers from Fig. 4.9

Trial[a,b,c]	Dense layer	Activation[d] layer and function	Batch[e] norm layer	Dropout layer	Training loss	Validation loss	Training accuracy	Validation accuracy	Validation precision (weighted)	Validation recall (weighted)	Validation F1-score (weighted)	Validation precision (macro)	Validation recall (macro)	Validation F1-score (macro)
a	1	2-ReLU	3	4	0.10	0.40	0.98	0.83	0.84	0.83	0.83	0.65	0.67	0.66
b	1	3-ReLU	2	4	0.06	0.40	0.99	0.85	0.85	0.85	0.85	0.67	0.68	0.68
c	1	4-ReLU	2	3	0.18	0.49	0.93	0.80	0.84	0.80	0.82	0.63	0.69	0.66
d	1	2-Sigmoid	3	-	0.25	0.41	0.90	0.82	0.86	0.82	0.84	0.66	0.73	0.69
e	1	2-Tanh	3	4	0.12	0.40	0.97	0.83	0.85	0.83	0.83	0.65	0.69	0.67
f	1	3-Tanh	2	4	0.13	0.49	0.96	0.79	0.83	0.79	0.81	0.60	0.64	0.62
g	1	4-Tanh	2	3	0.13	0.46	0.96	0.79	0.84	0.79	0.81	0.62	0.67	0.65
h	1	3-Sigmoid	2	4	0.59	0.58	0.81	0.84	0.86	0.74	0.78	0.62	0.73	0.67
i	1	2-Sigmoid	3	4	0.54	0.55	0.86	0.85	0.86	0.74	0.78	0.62	0.74	0.68

[a]The model with the optimal configuration parameters is shown in bold for this set of experiments. The order of layers is represent by the assigned number in the column's layer.

[b]A selection of experiments from Fig. 4.9 is reported for simplicity.

[c]All experiments have been standardised with the following hyperparamters: no. of input nodes = no. of input features, loss function = categorical cross-entropy, optimiser = Adam, learning rate = $1e^{-7}$, layer activation function = ReLU, output activation function = softmax, batch size = 512, dropout rate = 0.2, no. of output nodes = 2, no. of layers = 7, no. of nodes in hidden layers = 2048, dropout rate = 0.2, L2 penalty = 0.01.

[d]We test the three common activation functions for hidden layers: the rectified linear units (ReLU), the sigmoid, and the tanh activation functions (DeepAI, 2020; LeCun et al., 2012).

[e]We follow the definition by Ioffe & Szegedy (2015). Stack Overflow (2016), URL: https://stackoverflow.com/questions/39691902/ordering-of-batch-normalization-and-dropout.

4.4.2 Error backpropagation

The aim of a deep artificial neural network (ANN) is to evaluate the gradient[1] of the loss function[2]. We optimise the hyperparameters of the error backpropagation (BP) algorithm of the deep ANN, which is the iterative process of minimising the error function, i.e. improving the forecasting ability of the model (4.4.2.1). We run a set of experiments to determine the parameters of the loss function (4.4.2.2), followed by the learning rate experiment (4.4.2.3).

4.4.2.1 Error backpropagation algorithm

Most training algorithms involve an iterative procedure for minimising the error function, with weight adjustments made in sequences of steps[3] (Bishop, 2006). This is the optimisation function, which calculates the gradient, i.e. the partial derivative of the loss function with respect to weights, and the weights are modified in the opposite direction of the calculated gradient (Schraudolph & Cummins, 2006). One of the most well-known optimisation functions is the gradient descent (Ruder, 2017a). This iterative procedure is repeated until we reach the minima of the loss function[4].

We make use of the backpropagation algorithm for an arbitrary feed-forward network network with arbitrary non-linear activation functions, and a broad class of error functions. Specifically, the 7-layer deep ANN (4.4.1), is represented by Eqn. 4.9a, and in the generic form by Eqn. 4.9b (Lee et al., 2019). The error backpropagation algorithm forward propagates the input vector $\mathbf{x}_n$ through the activation functions. Then, it evaluates the error δ_k for each output k, which are the small differences in weights, and backpropagates them through the network[5] to evaluate the derivatives[6] (Bishop, 2006; Murphy, 2012; Nielsen, 2015). For the classification task (see 4.1), in which each input is assigned to one of K classes,

[1] Brownlee (2019i), URL: https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/.

[2] The loss function (also known as the cost function), $E(\mathbf{w})$ computes the error, i.e. the difference between the predictive value and the actual value, and which forward and back propagates through the ANN (Benvenuto & Piazza, 1992; Whittington & Bogacz, 2019).

[3] At each step, there are two distinct stages. The first stage involves the evaluation of the derivatives of the the error function with respect to the weights, which are back propagated through the network. In the second stage, the derivatives are used to forward compute the adjustments to be made to the weights. This is known as forward and backpropagation and is how the backpropagation algorithm computes the gradient of the loss function (Bishop, 2006; Murphy, 2012).

[4] Agrawal et al. (2017), URL: https://medium.com/data-science-group-iitr/loss-functions-and-optimization-algorithms-demystified-bb92daff331c.

[5] McGonagle et al. (2020), URL: https://brilliant.org/wiki/backpropagation/.

[6] For simplicity, the derivation of the backpropagation algorithm is omitted mainly due to the model complexity and the data complexity. The author refers to Bishop (2006), where a similar notation is used, and presents only the top level equations to help the readers understand the development of the methodology (Ng, 2017; Sadowski, 2017).

with the binary target variables $t_k \in \{0,1\}$, and with the network outputs interpreted as $y_k(\mathbf{x},\mathbf{w}) = p(t_k|\mathbf{x})$, the cross-entropy error function is described by Eqn. 4.9c, for each datapoint n in the training set (Bishop, 2006).

$$y_k(\mathbf{x}_n,\mathbf{w}) = h\left(\sum_{q=0}^{M} w_{rq}^{(7)}\sigma\left(\sum_{p=0}^{M} w_{qp}^{(6)}\sigma\left(\sum_{o=0}^{M} w_{po}^{(5)}\sigma\left(\sum_{m=0}^{M} w_{om}^{(4)}\sigma\left(\sum_{l=0}^{M} w_{ml}^{(3)}\sigma\left(\sum_{j=0}^{M} w_{lj}^{(2)}\sigma\left(\sum_{i=0}^{D} w_{ji}^{(1)}x_i\right)\right)\right)\right)\right)\right)\right) \tag{4.9a}$$

$$y_k(\mathbf{x}_n,\mathbf{w}) = h\left(\sum_{z=0}^{M} w_z\sigma_z(\mathbf{x})\right) \tag{4.9b}$$

$$E(\mathbf{w})_{CE} = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} log\left(y_k(\mathbf{x}_n,\mathbf{w})\right) \tag{4.9c}$$

where D is the number of nodes in the input layer, $z = 1,...,M$, and the subscript (1) indicates the corresponding parameters are in the first layer of the network, with same being applied to $\{i,j,l,m,o,p,q\}$, $\sigma(x) = sigmoid(x)$, $h(x) = softmax(x)$[1,2,3] are activation functions[4,5], $k = 1,...,K$ is the total number of outputs, CE = cross-entropy loss, and $n = 1,...,N$ is the number of datapoints.

There are many optimisation functions, which are also known as optimisers (Ruder, 2017a). We use the *Adam* (adaptive moment estimation)[6] optimiser because it is computationally efficient, easy to implement, and suitable for large scale data and parameter problems with noisy gradients[7]. This makes Adam one of the most popular optimisers in deep learning[8] (Bushaev, 2018).

[1]Roelants (2020), URL: https://peterroelants.github.io/posts/cross-entropy-softmax/

[2]Knet.jl (2020), URL: https://knet.readthedocs.io/en/latest/softmax.html

[3]Samala (2017), URL: https://becominghuman.ai/hierarchical-softmax-as-output-activation-function-in-neural-network-1d19089c4f49.

[4]Jain (2019), URL: https://www.linkedin.com/pulse/activation-functions-neural-networks-rahul-jain.

[5]Kathuria (2018), URL: https://blog.paperspace.com/vanishing-gradients-activation-function/.

[6]Adam is an extension of the classical stochastic gradient descent procedure to iteratively update network weights (Kingma & Ba, 2015). Stochastic gradient descent maintains a single learning rate for all weight updates and the learning rate does not change during training (Loshchilov & Hutter, 2019b; Ruder, 2017a). For Adam, a learning rate is maintained for each network weight (parameter) and separately adapted as learning unfolds. It computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients (Karpathy & Lei, 2015). Brownlee (2017b), URL: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.

[7]Adam combines the benefits of two other optimisers: the *adagrad* (adaptive gradient algorithm), which maintains a per-parameter learning rate improving performance on sparse gradient problems (LeCun et al., 2012), and *rmsprop* (root mean square propagation), which maintains a per-parameter learning rate adapted on the changing magnitude of the weight gradients (Qian, 1999).

[8]Karpathy (2017), URL: https://medium.com/@karpathy/a-peek-at-trends-in-machine-learning-ab8a1085a106.

4.4.2.2 Loss function

The loss function computes the ANN error between predicted and target value (Benvenuto & Piazza, 1992). We aim to determine the optimal hyperparameters for the loss function to converge to the minima, and thus identify the highest performing model. We adapt the cross-entropy loss function, defined by Eqn. 4.9c, into the focal loss function. The focal loss, developed by Facebook AI research, has weighted terms[1] in front of the cross-entropy loss to account for imbalanced datasets[2] (Haibo He & Garcia, 2009; Lin et al., 2017), and is defined by Eqn. 4.10.

$$E(\mathbf{w})_{FL} = -\sum_{n=1}^{N}\sum_{k=1}^{K} \alpha_k \Big(1 - y_k(\mathbf{x}_n, \mathbf{w})\Big)^{\gamma_k} t_{nk} log\Big(y_k(\mathbf{x}_n, \mathbf{w})\Big) \tag{4.10}$$

where α_k is a prefixed balancing value to balance the positive labelled and negative labelled samples (and one of the most common ways to balance the classes), γ_k is the focusing parameter, which down-weights the loss assigned to easily classified examples, $k = 1, ..., K$ is the total number of outputs, FL = focal cross-entropy loss, and $n = 1, ..., N$ is the number of datapoints.

Fig. 4.10[3] shows the loss function tuning experiments with focal cross-entropy loss, and is complemented by Table 4.7. From Fig. 4.10, we observe that the model's learning ability improves as the focusing parameters are varied. This is supported from the shape of downward shift of the training and validation loss curve (Brownlee, 2019h; Lin et al., 2017). The training and validation accuracy curves are shifted upwards and stability increases. We also observe that the validation precision (macro) increases, with the model becoming more sensitive to the data (Weber et al., 2019). From Table 4.7, we observe a stable learning for all models with no overfitting from the training and validation loss[4]. The optimal performance is exhibited by model f, with the highest validation F1-score (macro) (Mukhoti et al., 2020), which we use to determine the optimal learning rate (4.4.2.3).

[1] Du (2019), URL: https://medium.com/ai-salon/demystifying-focal-loss-i-a-more-focused-version-of-cross-entropy-loss-f49e4b044213.

[2] Focal loss down-weights the well-classified examples, instead of giving equal weighting to all training examples, This has the effect of putting more training emphasis on the data that is hard to classify. Kwag (2018), URL: https://chadrick-kwag.net/focal-loss-a-k-a-retinanet-paper-review/.

[3] Fig. 4.10 is a dynamic representation of the experiments, i.e. shows the evaluation of the different configuration models per epoch (iteration cycle) on the training and the validation datasets. It shows a selective number of experiments, which are overlayed due to the similarity in the results, and a sub-selection of those are reported in Table 4.7. Table 4.7 is a static representation of the experiments, i.e. shows the evaluation of the different configuration models at the last epoch (end of training) on the training and validation datasets.

[4] Nieradzik (2019), URL: https://lars76.github.io/neural-networks/object-detection/losses-for-segmentation/.

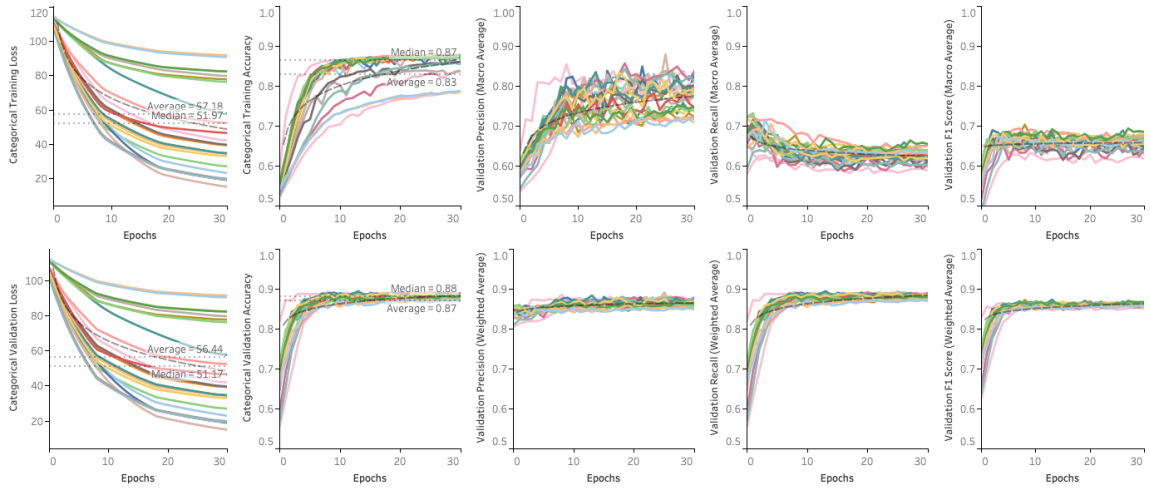


Fig. 4.10 Dynamic (per epoch) network loss function tuning evaluation results with focal cross-entropy loss for selective experiments: training loss, validation loss, validation precision (macro average), validation recall (macro average), validation F1-score (macro average), validation precision (weighted average), validation precision (weighted average), validation F1-score (weighted average)

Table 4.7 Static end of training (last epoch) network loss function tuning experiments with focal cross-entropy loss from Fig. 4.10

Trial[a,b,c]	α [d]	γ[e]	Weights[f]	Training loss	Validation loss	Training accuracy	Validation accuracy	Validation precision (weighted)	Validation recall (weighted)	Validation F1-score (weighted)	Validation precision (macro)	Validation recall (macro)	Validation F1-score (macro)
a	4.00	5.00	C,S	94.26	94.23	0.86	0.87	0.85	0.87	0.86	0.72	0.63	0.67
b	4.00	2.00	C,S	81.08	81.06	0.87	0.88	0.86	0.88	0.86	0.74	0.64	0.69
c	4.00	1.00	C,S	81.11	81.09	0.87	0.88	0.86	0.88	0.87	0.74	0.66	0.70
d	0.25	2.00	C	10.04	9.96	0.88	0.89	0.87	0.89	0.86	0.82	0.61	0.70
e	0.25	2.00	S	32.07	32.05	0.88	0.89	0.87	0.89	0.87	0.80	0.63	0.70
f	0.25	2.00	C,S	17.29	17.20	0.87	0.89	0.88	0.89	0.87	0.82	0.63	0.71

[a]The model with the optimal configuration parameters is shown in bold for this set of experiments. The order of layers is represent by the assigned number in the column's layer.

[b]A selection of experiments from Fig. 4.10 is reported for simplicity.

[c]All experiments have been standardised with the following hyperparamters: no. of input nodes = no. of input features, loss function = focal cross-entropy, optimiser = Adam, learning rate = $1e^{-7}$, layer activation function = ReLU, output activation function = softmax, batch size = 512, dropout rate = 0.2, no. of output nodes = 2, no. of layers = 7, no. of nodes in hidden layers = 2048, dropout rate = 0.2, L2 penalty = 0.01.

[d]Kapil (2018), URL: https://medium.com/adventures-with-deep-learning/focal-loss-demystified-c529277052de.

[e]Wei (2019), URL: https://www.dlology.com/blog/multi-class-classification-with-focal-loss-for-imbalanced-datasets/.

[f]There are two types of weights: class weights (C) and sample weights (S). Both weights involve the ratio of high-to-low value patents, with the difference of one being applied to the class level, and the other transforming the individual samples (Cui et al., 2019). Mao (2019), URL: https://leimao.github.io/blog/Focal-Loss-Explained/.

4.4.2.3 Learning rate

Deep learning ANN are trained using stochastic gradient descent optimisation algorithms (Bishop, 2006). The learning rate is a hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated (Bengio, 2012). We optimise the learning rate of the deep ANN to identify the optimal training time[1] required for the model to converge[2], since it controls how quickly the model is adapted to the task[3]. We use the Adam optimiser, an adaptive gradient descent algorithm, together with a learning scheduler, which monitors the performance of the model on the training dataset and the learning rate is adjusted in response[4] (Smith, 2017).

Fig. 4.11[5] shows the learning rate tuning experiments, and is complemented by Table 4.8. From Fig. 4.11, we observe that while the learning rate decreases, the training curve and validation curves converge closer together, with a smooth gradient. This is supported by an increase in the training accuracy, with a smooth learning, which is also replicated on the validation accuracy curve (Darken et al., 1992). The trend of the validation precision (macro) curve is slightly unstable, partly because it becomes more sensitive to the variation within the validation dataset. However, with decreasing learning rate, the validation precision curves (both macro and weighted) are shifted upwards, partly because the number of false positives (FP) decreases. This is also evident by Table 4.8, with also a slight increase in the validation recall (macro) due to a reduction in false negatives (FN) with the variation in the learning rate. We observe that ANN's learning momentum shifts towards a solution gradient, with the learning being slightly premature, i.e. stopped early since the training loss curve and validation loss curves do not converge to a flat trend line after 30 epochs (Qian, 1999; Smith, 2018; Smith & Topin, 2019). Table 4.8 model e is the model with the optimal performance, which we use for further evaluation in 4.5.2, and then test it with a variety of categorical output feature proxies in chapter 5.

[1]Choosing the learning rate is challenging as a value too small may result in a long training process that could get stuck, whereas a value too large may result in learning a sub-optimal set of weights too fast or an unstable training process.

[2]Lau (2017), URL:https://towardsdatascience.com/learning-rate-schedules-and-adaptive-learning-rate-methods-for-deep-learning-2c8f433990d1.

[3]Brownlee (2020h), URL: https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/.

[4]This is called an adaptive learning rate (Goodfellow et al., 2016).

[5]Fig. 4.11 is a dynamic representation of the experiments, i.e. shows the evaluation of the different configuration models per epoch (iteration cycle) on the training and the validation datasets. It shows a selective number of experiments, which are overlayed due to the similarity in the results, and a sub-selection of those are reported in Table 4.8. Table 4.8 is a static representation of the experiments, i.e. shows the evaluation of the different configuration models at the last epoch (end of training) on the training and validation datasets.

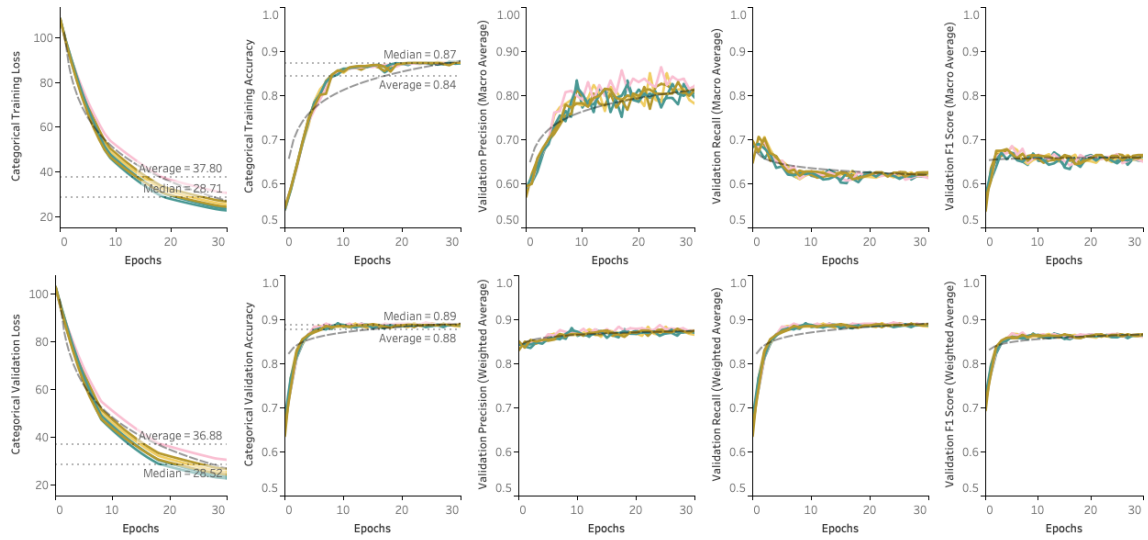


Fig. 4.11 Dynamic (per epoch) network learning rate tuning evaluation results for selective experiments: training loss, validation loss, validation precision (macro average), validation recall (macro average), validation F1-score (macro average), validation precision (weighted average), validation precision (weighted average), validation F1-score (weighted average)

Table 4.8 Network tuning experiments with learning rate variation from Fig. 4.11

Trial[a,b,c]	Learning rate[d]	Training loss[e]	Validation loss	Training accuracy	Validation accuracy	Validation precision (weighted)	Validation recall (weighted)	Validation F1-score (weighted)	Validation precision (macro)	Validation recall (macro)	Validation F1-score (macro)
a	$1e^{-6}$	25.81	25.68	0.88	0.89	0.87	0.89	0.86	0.79	0.62	0.69
b	$1e^{-8}$	23.05	22.93	0.88	0.89	0.87	0.89	0.86	0.80	0.62	0.70
c	$5e^{-7}$	24.72	24.60	0.88	0.89	0.87	0.89	0.86	0.80	0.62	0.70
d	$3e^{-7}$	23.92	23.79	0.87	0.89	0.87	0.89	0.86	0.81	0.62	0.70
e	$1e^{-7}$	27.07	26.93	0.87	0.89	0.88	0.89	0.87	0.81	0.63	0.71

[a] The model with the optimal configuration parameters is shown in bold for this set of experiments.

[b] A selection of experiments from Fig. 4.11 is reported for simplicity.

[c] All experiments have been standardised with the following hyperparamters: no. of input nodes = no. of input features, loss function = focal cross-entropy, optimiser = Adam, layer activation function = sigmoid, output activation function = softmax, batch size = 512, dropout rate = 0.2, no. of output nodes = 2, no. of layers = 7, no. of nodes in hidden layers = 2048, dropout rate = 0.2, L2 penalty = 0.01.

[d] We vary the learning rate and monitor the performance of the model. Brownlee (2016c), URL: https://machinelearningmastery.com/using-learning-rate-schedules-deep-learning-models-python-keras/.

[e] The evaluation metrics are introduced in 4.3 and the dataset terminology of training, validation and testing is introduced in 4.5.1.

4.5 Deep neural network architecture implementation

In 4.5 we describe the implementation tests for the development and deployment of the deep and wide ANN. The aim is to ensure that we evaluate our developed deep learning algorithmic approach on suitable representations of the dataset, to ensure the robustness and reliability of our proposed approach[1]. Thus, section 4.5 provides support for the development of the deep learning algorithmic approach. We follow a 2-step evaluation strategy: (i) we describe the dataset processing, dataset variations and dataset split into training, validation and testing datasets, to evaluate the performance of our models (4.5.1); (ii) we describe the evaluation strategies for different variations of the dataset for training, validation and testing of our models, including the out-of-sample (OOS), out-of-time (OOT) evaluation tests and the cross validation tests for assessing the model's generalising ability (4.5.2).

4.5.1 Dataset split

We evaluate AI methods, and specifically supervised paradigm models on a dataset, which is split into training, validation and testing dataset. The purpose of this is to ensure: (i) the models are able to work with data that have not been exposed to before[2,3], and (ii) the datasets have a suitable representation of the classification task, i.e. the distribution of the categoric output/ target proxies is the same in the training, validation and testing datasets[4].

The training dataset is defined as the part of data used to fit the model, i.e. a set of examples used for learning and fit the parameters of the classifier. The validation dataset is defined as the part of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The testing dataset[5] is defined as the part of data used to provide an unbiased evaluation of a final model fit on the training dataset, i.e. to assess the performance of a fully-specified classifier (Bishop, 2006). There are some other techniques, complementary to the train-validate-test split[6]

[1] Ruder (2017b), URL: https://ruder.io/transfer-learning/.

[2] The model should be evaluated on datapoints that are not used to build or fine-tune the model, i.e. they are not part of the training or validation datasets. So that they provide an unbiased sense of model effectiveness (Kuhn & Johnson, 2013; Russell & Norvig, 2010).

[3] Kumar (2020), URL: https://towardsdatascience.com/data-splitting-technique-to-fit-any-machine-learning-model-c0d7f3f1c790.

[4] Seif (2018), URL: https://towardsdatascience.com/handling-imbalanced-datasets-in-deep-learning-f48407a0e758.

[5] Brownlee (2017g), URL: https://machinelearningmastery.com/difference-test-validation-datasets/.

[6] Brownlee (2020g), URL: https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/.

above, of calculating unbiased estimates of a model's generalising ability. These include k-fold and random split cross validation to tune the model's hyperparameters (4.5.2.2).

We follow a mixed approach, which builds on the train-test split approach and cross validation approach, with variable percentage splits for the training, validation and training datasets[1,2]. Fig. 4.12 shows the variations of the dataset and how the dataset is constrained according to the time horizon and the output proxy. For every output proxy, we ensure that the dataset is constrained to only the fields that contain the full features, i.e. there is a complete set of input feature determinants and output proxies for every patent, and there are no NaNs (empty fields)[3].

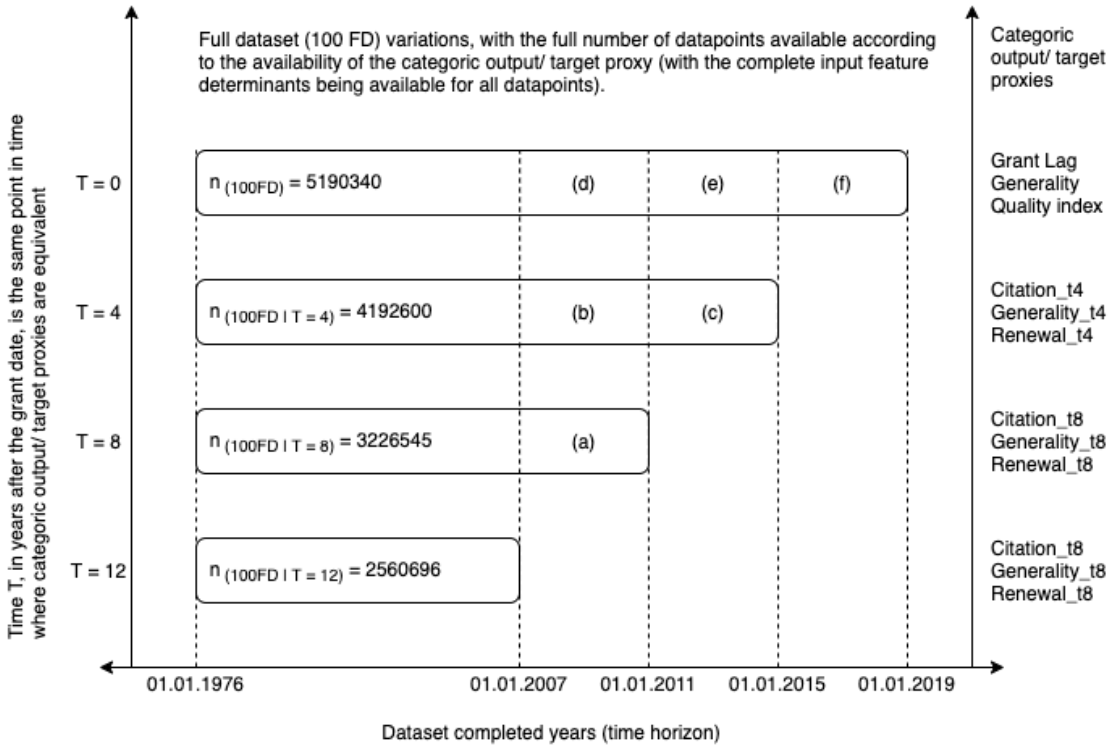


Fig. 4.12 Dataset variations with full datapoints for different time horizons and output proxies

Fig. 4.13 shows the different evaluation strategies we follow with the percentage split of the dataset, associated output proxy and reference table (Dobbin & Simon, 2011). Setting up the training, validation and testing datasets has an impact on the evaluation of the generalising

[1]Sanjay (2018), URL: https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f.

[2]Bronshtein (2017), URL: https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6.

[3]For example, for the time horizon T=4, for the output proxies of citations_t4, generality_t4 and renewal_t4, there are 4192600 available datapoints with completed fields.

ability of the model. It is important to choose the validation and testing datasets from the same distribution as the training dataset for the output proxy. In addition, the size of the validation and testing datasets depend on the size of the available dataset to assess the performance of the model[1,2]. We perform the dataset split breakdown in our evaluation strategies: (i) the cross validation in 4.5.2.2 with a 70:20:10 percentage split of the training/ validation/ testing datasets, (ii) train and test our models on the full dataset (100FD), the 10% sample dataset (010FD), and the 3% sample dataset (003FD) with a 98:1:1 percentage because the size of the dataset is very large (Xu & Goodacre, 2018).

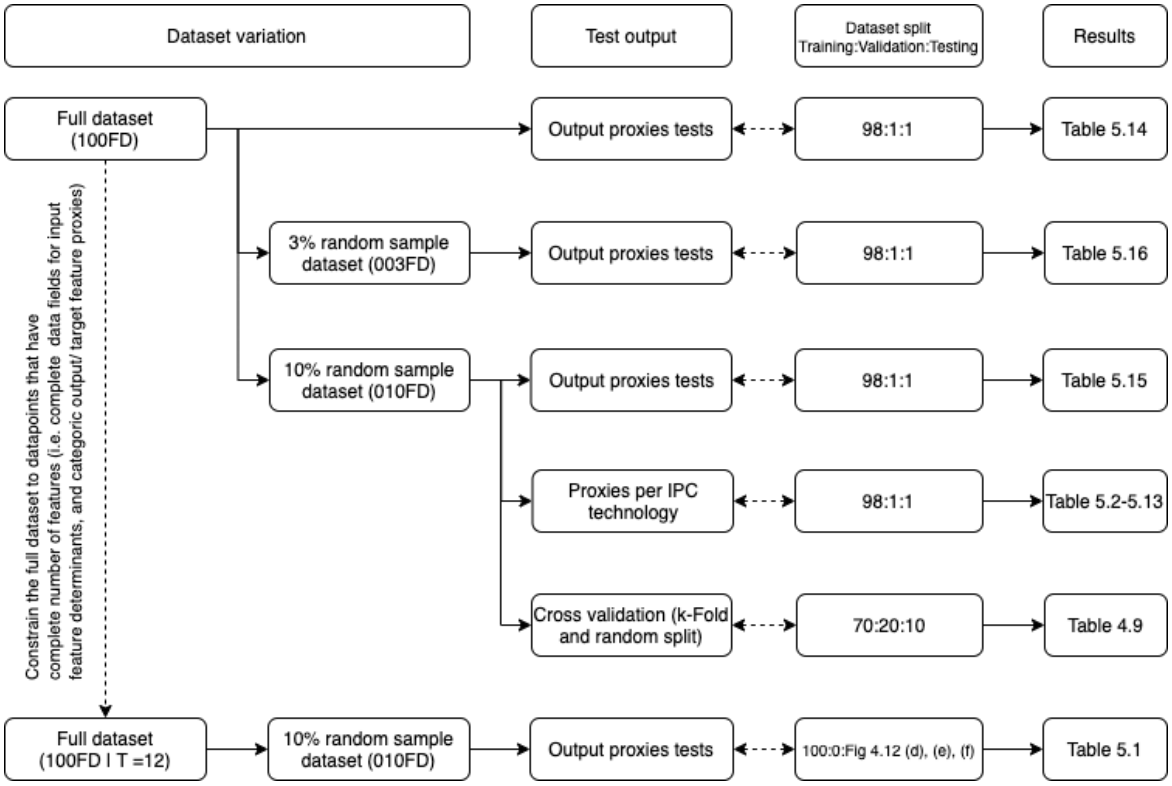


Fig. 4.13 Evaluation strategies for training, validation and testing, based on the train-test split approach and cross validation approach, with the associated results table (see 4.5.2)

4.5.2 Evaluation strategies for training, validation and testing

We evaluate our developed deep learning approach on suitable representations of the dataset, to ensure robustness and reliability of our proposed approach. We use the evaluation metrics explained in 4.3 to evaluate the models on the testing datasets. We follow an evaluation

[1]Ng & Katanforoosh (2020), URL: http://cs230.stanford.edu/blog/split/.

[2]Kumar (2020), URL: https://towardsdatascience.com/data-splitting-technique-to-fit-any-machine-learning-model-c0d7f3f1c790.

approach with variable dataset splits and variable sample sizes (see Fig. 4.13). The aim of this evaluation approach is to ensure robustness and reliability of our proposed approach by performing the following: (i) out-of-sample (OOS) tests[1], (ii) out-of-time (OOT) tests[2], and (iii) cross validation[3].

4.5.2.1 Evaluation strategies breakdown

We perform the following evaluations (Wu et al., 2012a), shown in Fig. 4.13: (i) an OOS test on the full dataset (100FD) for all output proxies (see 3.3) with a dataset split of 98:1:1 for training:validation:testing; (ii) an OOS test on a 10% random representative sample dataset (010FD)[4] for all output proxies with a dataset split of 98:1:1 for training:validation:testing[5]; (iii) an OOS test on a 10% random representative sample dataset (010FD) for all output proxies constrained to technology IPC classification classes with a dataset split of 98:1:1 for training:validation:testing; (iv) an OOS test on a 3% random representative sample dataset (003FD)[6] for all output proxies with a dataset split of 98:1:1 for training:validation:testing[7]; (v) a cross validation test (k-Fold[8] and random split[9]) on the 10% random representative

[1]An out-of-sample (OOS) test, is a forecasting test conducted, when a model is tested on a holdout (previously unseen) testing dataset. The test is used to assess the ability of the model to forecast known values, i.e. the testing dataset. The testing dataset is a percentage of a original dataset, which is not used for training and validation (Beleites et al., 2013; Bergdahl et al., 2007; Tashman, 2000).

[2]An out-of-time (OOT) test is an extension of the out-of-sample (OOS) test, when a model is tested on a holdout (previously unseen) testing dataset, which is not a percentage of the original dataset. The testing dataset is an extension of the original population of the full dataset, because of new observations (see Fig. 4.12 and 4.13) (Beleites et al., 2013; Bergdahl et al., 2007; Tashman, 2000).

[3]Schneider & Moore (1997), URL: https://www.cs.cmu.edu/~schneide/tut5/node42.html.

[4]The 10% random representative sample dataset (010FD) is a 10% random sample of the full dataset (100FD), where year distribution and IPC distribution of patents have been stratified (Brownlee, 2017g, 2020g; Dobbin & Simon, 2011; Ng & Katanforoosh, 2020).

[5]For each output proxy, the 010FD sample dataset is split into a training dataset (98%), validation dataset (1%) and testing dataset (1%), ensuring a representative distribution of the categoric output proxy in the training, validation and testing datasets (Ng & Katanforoosh, 2020).

[6]The 3% random representative sample dataset (003FD) is a 3% random sample of the full dataset (100FD), where year distribution and IPC distribution of patents have been stratified (Brownlee, 2017g, 2020g; Dobbin & Simon, 2011; Ng & Katanforoosh, 2020).

[7]For each output proxy, the 003FD sample dataset is split into a training dataset (98%), validation dataset (1%) and testing dataset (1%), ensuring a representative distribution of the categoric output proxy in the training, validation and testing datasets (Ng & Katanforoosh, 2020).

[8]k-Fold cross validation (k = 10) is when a dataset is split between a training/ validation dataset (90%) and an out-of-sample test set (10%). A model is trained on (k-1) number of folds and validated on a 1 fold of the training/ validation (90%) dataset, and then tested on an out-of-sample test (10%) dataset (Kohavi, 1995). Pedregosa et al. (2019), URL: https://scikit-learn.org/stable/auto_examples/model_selection/plot_cv_indices.html#sphx-glr-auto-examples-model-selection-plot-cv-indices-py.

[9]Random split (split = 10) is when the dataset is split into arrays or matrices into random train and test subsets. We calculate the average of the performance of the model across each split, which will give a better estimate of the generalising ability of the model.

sample dataset (010FD) for all output proxies with a dataset split of 70:20:10 for training/ validation/ testing; (vi) an OOT test on a 10% random sample (010FD | T=12), of a a subset of the full dataset (100FD | T=12)[1]. The trained and validated models are then tested on a random 20% OOT sample of all available output proxies from 2007-2019[2,3]. The results of these tests are transparently presented in chapter 5.

4.5.2.2 Cross validation (k-Fold and random split)

Table 4.9 shows the results of the cross validation evaluation on the 010FD dataset, with a random split and a k-Fold[4] cross validation (Schneider & Moore, 1997). Column a represent the validated accuracy on that fold/ split, column b represent the model being tested on the out-of-sample test set, and column c represent an ensemble[5] of models combined together and tested on the holdout test. Ensemble learning[6,7,8] are methods that combine the predictions from multiple models.

We observe that for all splits and folds the standard deviation is low between 0.001-0.002, indicating that the cross validation method is effective with the distribution of the output proxies in the training, validation and testing datasets[9]. This ensures that the distribution of the proxies is representative in all datasets (see 4.1). We observe a stability in the models for all proxies for columns a and b, and for proxies such as generality_t8 (column c) the ensemble performs slightly better. The k-Fold cross validation appears less optimistic than the random split[10]. From all the results in Table 4.9, we observe that the accuracy of the proposed deep learning method is stable for all output proxies.

[1]This subset consists of all granted patents with complete fields of features, i.e. all the outputs in the time frame T=12 exist (see Fig. 4.13). This means that the patents have reached aged 12, which constrains the dataset to the years between 1976-2007, and we take a 10% random representative sample from that subset to form the sample dataset (010FD | T=12)

[2]All output proxies from 2007-2019 are divided into three time subsets of the full dataset (100FD): (i) between 2007-2011 (equivalent to Fig. 4.12 part d), (ii) between 2011-2015 (equivalent to Fig. 4.12 part e), and (iii) between 2015-2019 (equivalent to Fig. 4.12 part f).

[3]This is both an out-of-sample test, i.e. the models have never 'seen' the test dataset, and an out-of-time test, i.e. the models are trained and validated up to 2007, and then forecast in the future from 2007-2019.

[4]Brownlee (2018b), URL: https://machinelearningmastery.com/k-fold-cross-validation/.

[5]Shubham (2018), URL: https://becominghuman.ai/ensemble-learning-bagging-and-boosting-d20f38be9b1e.

[6]Rocca (2019), URL: https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205.

[7]Brownlee (2018c), URL: https://machinelearningmastery.com/a-gentle-introduction-to-the-bootstrap-method/.

[8]Brownlee (2017a), URL:https://machinelearningmastery.com/transfer-learning-for-deep-learning/.

[9]Sanjay (2018), URL: https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f.

[10]Brownlee (2018e), URL: https://machinelearningmastery.com/how-to-create-a-random-split-cross-validation-and-bagging-ensemble-for-deep-learning-in-keras/.

Table 4.9 Cross validation (Random Split and k-Fold)

Output[a]			Overall[b]									Short Term (t4)									Medium Term (t8)									Long Term (t12)								
			Grant Lag			Generality			Quality Index 4			Citations_t4			Generality_t4			Renewal_t4			Citations_t8			Generality_t8			Renewal_t8			Citations_t12			Generality_t12			Renewal_t12		
Datapoints[c]	Class		V_H		V_L	V_H	V_L		V_H		V_L	V_H		V_L	V_H		V_L	V_H		V_L	V_H		V_L	V_H		V_L	V_H		V_L	V_H		V_L	V_H		V_L	V_H		V_L
	Total		129598		389431	69174		362916	42658		476376	4015		415426	9612		409829	361519		57664	21202		301497	18906		303793	209474		119676	37498		244183	22836		233237	113582		165675
	Train / Val		116638		350488	62257		326624	38392		428738	3613		373883	8651		368845	325366		51898	19082		271347	17015		273414	188527		107708	34089		221984	20552		209913	102224		149107
	Testing		12960		38943	6917		36292	4266		47638	402		41543	961		40984	36153		5766	2120		30150	1891		30379	20947		11968	3409		22199	2284		23324	11358		16568
Random Split (10 splits)[d]																																						
Columns			a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c
Accuracy	Split	1	0.753	0.753	0.753	0.844	0.844	0.844	0.943	0.943	0.943	0.989	0.989	0.989	0.976	0.977	0.977	0.859	0.858	0.858	0.932	0.932	0.932	0.939	0.938	0.938	0.662	0.663	0.663	0.875	0.876	0.876	0.905	0.905	0.905	0.649	0.649	0.649
		2	0.753	0.754	0.753	0.844	0.845	0.846	0.942	0.942	0.943	0.989	0.989	0.989	0.976	0.976	0.977	0.860	0.859	0.858	0.931	0.931	0.932	0.939	0.939	0.939	0.664	0.663	0.664	0.875	0.875	0.876	0.905	0.904	0.905	0.650	0.652	0.651
		3	0.754	0.754	0.754	0.844	0.845	0.846	0.942	0.942	0.942	0.989	0.989	0.989	0.976	0.977	0.977	0.859	0.858	0.859	0.931	0.930	0.931	0.939	0.939	0.939	0.662	0.665	0.665	0.875	0.877	0.876	0.905	0.905	0.905	0.649	0.649	0.651
		4	0.753	0.753	0.755	0.842	0.844	0.846	0.943	0.943	0.943	0.989	0.989	0.989	0.976	0.977	0.977	0.859	0.858	0.859	0.931	0.931	0.931	0.939	0.938	0.939	0.663	0.662	0.665	0.875	0.875	0.876	0.905	0.904	0.905	0.651	0.650	0.651
		5	0.752	0.752	0.754	0.842	0.844	0.846	0.943	0.942	0.943	0.989	0.989	0.989	0.976	0.976	0.977	0.859	0.858	0.858	0.931	0.931	0.931	0.939	0.938	0.939	0.662	0.665	0.666	0.875	0.875	0.876	0.905	0.904	0.905	0.652	0.652	0.652
		6	0.751	0.752	0.755	0.843	0.844	0.846	0.943	0.943	0.943	0.989	0.989	0.989	0.976	0.977	0.977	0.859	0.859	0.859	0.931	0.931	0.931	0.939	0.938	0.939	0.660	0.664	0.665	0.875	0.875	0.876	0.906	0.905	0.905	0.650	0.650	0.652
		7	0.751	0.753	0.755	0.845	0.846	0.846	0.943	0.943	0.943	0.989	0.989	0.989	0.976	0.976	0.977	0.859	0.859	0.859	0.931	0.931	0.931	0.939	0.938	0.939	0.662	0.664	0.666	0.875	0.876	0.876	0.905	0.904	0.905	0.649	0.652	0.652
		8	0.752	0.753	0.755	0.844	0.845	0.846	0.943	0.943	0.943	0.989	0.989	0.989	0.976	0.977	0.977	0.860	0.859	0.859	0.932	0.931	0.931	0.939	0.938	0.939	0.662	0.662	0.665	0.875	0.876	0.876	0.904	0.904	0.905	0.648	0.651	0.653
		9	0.753	0.754	0.755	0.844	0.845	0.846	0.943	0.942	0.943	0.989	0.989	0.989	0.976	0.976	0.977	0.860	0.858	0.859	0.932	0.931	0.931	0.939	0.938	0.939	0.662	0.665	0.666	0.875	0.876	0.876	0.905	0.904	0.905	0.651	0.650	0.652
		10	0.753	0.753	0.755	0.844	0.844	0.846	0.942	0.942	0.943	0.989	0.989	0.989	0.976	0.976	0.977	0.860	0.859	0.859	0.932	0.932	0.931	0.939	0.938	0.939	0.663	0.665	0.666	0.875	0.874	0.876	0.905	0.904	0.905	0.649	0.652	0.653
Average Accuracy			0.753	0.753	0.754	0.844	0.845	0.846	0.943	0.943	0.943	0.989	0.989	0.989	0.976	0.977	0.977	0.859	0.859	0.859	0.931	0.931	0.931	0.939	0.938	0.939	0.662	0.664	0.665	0.875	0.876	0.876	0.905	0.904	0.905	0.650	0.651	0.652
Standard Deviation			0.001	0.001	0.001	0.001	0.001	0.001	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.001	0.001	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.001	0.001	0.001	0.000	0.001	0.000	0.000	0.000	0.000	0.001	0.001	0.001
k-Fold cross validation (k = 10)[e]																																						
Columns[f]			a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c
Accuracy	Fold	1	0.755	0.755	0.755	0.846	0.846	0.846	0.944	0.943	0.943	0.989	0.989	0.989	0.976	0.976	0.976	0.858	0.858	0.858	0.931	0.933	0.933	0.939	0.939	0.939	0.660	0.665	0.665	0.879	0.876	0.876	0.906	0.904	0.904	0.652	0.652	0.652
		2	0.753	0.755	0.755	0.842	0.845	0.846	0.941	0.943	0.943	0.990	0.989	0.989	0.977	0.976	0.976	0.860	0.858	0.858	0.932	0.933	0.933	0.938	0.938	0.939	0.659	0.665	0.666	0.873	0.870	0.876	0.907	0.905	0.904	0.651	0.650	0.653
		3	0.752	0.752	0.755	0.849	0.846	0.846	0.943	0.943	0.943	0.989	0.989	0.989	0.975	0.976	0.976	0.859	0.858	0.858	0.935	0.933	0.933	0.938	0.938	0.939	0.663	0.666	0.667	0.878	0.875	0.876	0.905	0.905	0.904	0.647	0.655	0.654
		4	0.754	0.753	0.755	0.846	0.846	0.846	0.942	0.943	0.943	0.990	0.989	0.989	0.976	0.976	0.976	0.859	0.858	0.858	0.932	0.932	0.933	0.938	0.938	0.939	0.662	0.664	0.667	0.878	0.876	0.876	0.904	0.905	0.905	0.652	0.651	0.655
		5	0.752	0.755	0.755	0.843	0.847	0.846	0.943	0.943	0.943	0.989	0.989	0.989	0.976	0.976	0.976	0.860	0.858	0.858	0.932	0.932	0.933	0.939	0.938	0.939	0.658	0.665	0.667	0.873	0.871	0.875	0.908	0.904	0.905	0.649	0.651	0.655
		6	0.753	0.754	0.756	0.850	0.847	0.847	0.941	0.943	0.943	0.990	0.989	0.989	0.976	0.976	0.976	0.858	0.858	0.858	0.932	0.932	0.932	0.940	0.938	0.939	0.662	0.663	0.666	0.873	0.872	0.876	0.902	0.905	0.905	0.648	0.650	0.654
		7	0.757	0.754	0.756	0.843	0.846	0.847	0.942	0.943	0.943	0.989	0.989	0.989	0.976	0.976	0.976	0.861	0.858	0.858	0.934	0.932	0.933	0.937	0.938	0.939	0.662	0.665	0.667	0.881	0.875	0.872	0.903	0.904	0.905	0.650	0.652	0.654
		8	0.751	0.754	0.755	0.846	0.845	0.846	0.943	0.943	0.943	0.988	0.989	0.989	0.976	0.977	0.976	0.860	0.858	0.858	0.933	0.931	0.933	0.935	0.938	0.939	0.660	0.663	0.667	0.877	0.874	0.874	0.906	0.905	0.905	0.644	0.651	0.654
		9	0.751	0.754	0.756	0.845	0.845	0.846	0.942	0.943	0.943	0.989	0.989	0.989	0.976	0.976	0.976	0.860	0.858	0.858	0.931	0.932	0.933	0.939	0.938	0.939	0.662	0.666	0.667	0.877	0.870	0.872	0.907	0.905	0.905	0.654	0.652	0.654
		10	0.758	0.754	0.755	0.844	0.845	0.847	0.943	0.943	0.943	0.988	0.989	0.989	0.975	0.976	0.976	0.861	0.858	0.858	0.931	0.932	0.933	0.937	0.938	0.939	0.665	0.665	0.667	0.877	0.874	0.874	0.905	0.905	0.905	0.656	0.650	0.654
Average Accuracy			0.754	0.754	0.755	0.845	0.846	0.846	0.942	0.943	0.943	0.989	0.989	0.989	0.976	0.976	0.976	0.860	0.858	0.858	0.932	0.932	0.933	0.938	0.938	0.939	0.661	0.665	0.667	0.877	0.873	0.875	0.905	0.905	0.905	0.650	0.651	0.654
Standard deviation			0.002	0.001	0.000	0.003	0.001	0.000	0.001	0.000	0.000	0.001	0.000	0.000	0.001	0.000	0.000	0.001	0.000	0.000	0.001	0.001	0.000	0.001	0.000	0.000	0.002	0.001	0.001	0.003	0.002	0.002	0.002	0.000	0.000	0.003	0.001	0.001

[a] All outputs are defined according to Table 3.2. Class V_H represent high value patents and class V_L represent low value patents. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] Outputs in the overall category are constructed and calculated according to Squicciarini et al. (2013).

[c] The random split and k-fold cross validation are calculated on the 10% random sample of the full dataset (where year distribution and IPC distributions have been stratified). The 10% random sample dataset is split into three sets (maintaining the distribution of the output variable in each): training, validation and the testing set, with a ratio of 70:20:10 respectively.

[d] Random split (split = 10) is when the dataset is split into arrays or matrices into random train and test subsets. We calculate the average of the performance of the model across each split, which will give a better estimate of the generalising ability of the model.

[e] k-Fold Cross validation (k=10) is when a dataset is split between a training/ validation dataset (90%) and an out-of-sample test set (10%); a model is trained on (k-1) number of folds and validated on a 1 fold of the training/ validation dataset, and then tested on an out-of-sample test set (Kohavi, 1995). Pedregosa et al. (2019), URL: https://scikit-learn.org/stable/auto_examples/model_selection/plot_cv_indices.html#sphx-glr-auto-examples-model-selection-plot-cv-indices-py.

[f] Columns a represent the validated accuracy on that fold, i.e. an individual model is trained on folds 2-9 and validated on fold 1. Columns b represent the model being tested on the out-of-sample test set. Columns c represent an ensemble of models combined together and tested on the holdout test, i.e. ensemble 3 consists of model 1 (trained on folds 2-9, validated on fold 1) and model 2 (trained on folds 1, 3-9, validated on fold 2) and model 3 (trained on folds, 1, 2, 4-9, validated on fold 3, combined together and tested on the out-of-sample test set.

Chapter 5

Empirical Results

In this chapter, we describe the results from the deployment of the deep learning algorithmic approach (chapter 4), using our developed dataset (chapter 3). Class V_H represents high value patents and class V_L represents low value patents (see 3.2.2.3 and Tables 3.2 and 4.1).

We aim to evaluate our developed deep learning approach on suitable representations of the dataset, and ensure robustness and reliability of our proposed approach. We use the evaluation metrics explained in 4.3 to evaluate the models on the testing datasets. We present the results on the testing datasets from three evaluation strategies (see 4.5 and Fig. 4.13): (i) the out-of-sample (OOS)[1] and out-of-time (OOT)[2] evaluation strategy (5.1); (ii) the out-of-sample (OOS) by technological area (5.2); and (iii) the out-of-sample (OOS) by sample size evaluation strategy (5.3). We describe the results and provide some explanations of the observed trends. The purpose is to explain some of the observations, which arise from the analysis of using a transparent reporting structure and multiple evaluation metrics. The discussion and implications are found in chapter 6, where we draw observations from multiple results tables and compare them collectively to the literature (chapter 2).

[1] An out-of-sample (OOS) test, is a forecasting test conducted, when a model is tested on a holdout (previously unseen) testing dataset. The test is used to assess the ability of the model to forecast known values, i.e. the testing dataset. The testing dataset is a percentage of a original dataset, which is not used for training and validation (Beleites et al., 2013; Bergdahl et al., 2007; Tashman, 2000).

[2] An out-of-time (OOT) test is an extension of the out-of-sample (OOS) test, when a model is tested on a holdout (previously unseen) testing dataset, which is not a percentage of the original dataset. The testing dataset is an extension of the original population of the full dataset, because of new observations (see Fig. 4.12 and 4.13) (Beleites et al., 2013; Bergdahl et al., 2007; Tashman, 2000).

5.1 Model out-of-sample evaluation by an out-of-time evaluation strategy

Table 5.1 shows the results of the out-of-sample and out-of-time test for all outputs, following the evaluation strategy in 4.5[1].

For the overall outputs, i.e. grant lag, generality and quality_index_4, for the 2007-2011 test dataset, we observe that the quality_index_4 (model c) has the highest accuracy with 0.94, and macro average F1-score with 0.80, similar to Table 5.15 results. However, the false negative rate (FNR) for model a (grant_lag) is 0.40, driven by the high number of false negatives. This indicates that characteristics of the V_H and V_L patents are not fully identified or several patents are granted at adhoc times distorting the output proxy. This seems to be supported by Squicciarini et al. (2013), which identify a number of peaks in the publication life cycle of patents from year 2004 onwards, suggesting the existence of administrative rules shaping the timing of grant of patents. For the 2011-2015 test dataset and the 2015-2019, the grant_lag output improves even further to accuracies of 0.75 (model m) and 0.87 (model y) respectively, and macro average F1-scores of 0.77 and 0.87 respectively. The reasons behind this are threefold: (i) a substantial improvement in the digital landscape, reducing the search times, and administrative burden, and thus improving search and examination times (Squicciarini et al., 2013); (ii) the changing competitive landscape has forced firms to be more precise with their patent and look for a quick and robust granting process of the V_H patents, indicated also by the rising precision of the V_H patents from 0.18 (model a), 0.40 (model m), to 0.91 (model y) (Harhoff et al., 2003; Thoma, 2014); (iii) the distribution of V_H patents is higher in later test subsets, driven by the desire for patents to be granted faster (Harhoff et al., 2007).

For the citations outputs, i.e. citations_t4, citations_t8, and citations_t12, we observe a reduction in accuracy from 0.99 for citation_t4 (model d), to 0.93 for citation_t8 (model e), to 0.88 for citation_t12 (model f), for 2007-2011. This is consistent with Table 5.15, with an improvement of the V_H patents recall, and a marginal increase in the false negative rate.

[1]The models are trained and validated on a 10% random sample (010FD | T=12), of a a subset of the full dataset (100FD | T=12). This subset consists of all granted patents with complete fields of features, i.e. all the outputs in the time frame T=12 exist (see Fig. 4.13). This means that the patents have reached aged 12. This constrains the dataset to the years between 1976-2007, and we take a 10% random representative sample from that subset to form the sample dataset (010FD | T=12). These trained and validated models are then tested on a random 20% out-of-time sample of all available output proxies from 2007-2019, in 3 time sub-datasets of the full dataset (100FD): (i) between 2007-2011 (equivalent to Fig. 4.12 part d), (ii) between 2011-2015 (equivalent to Fig. 4.12 part e), and (iii) between 2015-2019 (equivalent to Fig. 4.12 part f). Therefore, this is both an out-of-sample test, i.e. the models have never 'seen' the test dataset, and an out-of-time test, i.e. the models are trained and validated up to 2007, and then forecast in the future from 2007-2019.

Table 5.1 Model Evaluation Out of Time Test

Output[a,b]			Overall			Citations			Generality			Renewals		
			Grant_Lag	Generality	Quality_Index_4	Citations_t4	Citations_t8	Citations_t12	Generality_t4	Generality_t8	Generality_t12	Renewals_t4	Renewals_t8	Renewals_t12
2007-2011														
Model			a	b	c	d	e	f	g	h	i	j	k	l
Datapoints	Class[c]	V_H	16285	42111	22573	2437	15152	2168	6587	12922	1489	211215	98509	25999
		V_L	184734	182785	228701	188692	213992	17218	232325	216222	17897	32823	58223	239097
Accuracy[d]			0.63	0.82	0.94	0.99	0.93	0.88	0.97	0.94	0.92	0.86	0.64	0.56
Precision	Class	V_H	0.18	0.59	0.80	0.44	0.42	0.39	0.08	0.40	0.38	0.87	0.65	0.93
		V_L	1.00	0.84	0.95	0.99	0.94	0.91	0.97	0.95	0.93	0.36	0.54	0.13
	Average	Macro	0.59	0.72	0.88	0.72	0.68	0.65	0.53	0.68	0.66	0.62	0.60	0.53
		Weighted	0.93	0.79	0.94	0.98	0.91	0.85	0.95	0.91	0.89	0.80	0.61	0.85
Recall	Class	V_H	0.99	0.19	0.48	0.01	0.11	0.20	0.01	0.04	0.08	1.00	0.91	0.56
		V_L	0.60	0.97	0.99	1.00	0.99	0.96	1.00	1.00	0.99	0.02	0.19	0.62
	Average	Macro	0.79	0.58	0.73	0.51	0.55	0.58	0.50	0.52	0.54	0.51	0.55	0.59
		Weighted	0.63	0.82	0.94	0.99	0.93	0.88	0.97	0.94	0.92	0.86	0.64	0.56
F1-score	Class	V_H	0.30	0.29	0.60	0.02	0.17	0.26	0.01	0.07	0.14	0.93	0.76	0.70
		V_L	0.75	0.90	0.97	0.99	0.96	0.93	0.97	0.97	0.96	0.03	0.28	0.22
	Average	Macro	0.68	0.64	0.80	0.60	0.61	0.61	0.51	0.59	0.59	0.56	0.57	0.56
		Weighted	0.71	0.79	0.94	0.98	0.91	0.86	0.96	0.92	0.89	0.81	0.58	0.65
Confusion	True Positives (TP)		111011	177342	226066	188657	211717	16551	232314	215479	17697	210236	89207	16101
Matrix	False Positives (FP)		232	34117	11723	2409	13509	1737	6596	12420	1364	32267	47221	105907
	False Negatives (FN)		73723	5443	2635	35	2275	667	11	743	200	979	9302	9898
	True Negatives (TN)		16053	7994	10850	28	1643	431	1	502	125	556	11002	133190
False Negative Rate (FNR)			0.40	0.03	0.01	0.00	0.01	0.04	0.00	0.00	0.01	0.00	0.09	0.38
Mean Absolute Error (MAE)			0.37	0.17	0.06	0.01	0.07	0.12	0.03	0.06	0.08	0.14	0.36	0.44
2011-2015														
Model			m	n	o	p	q	r	s	t	u	v	w	x
Datapoints	Class	V_H	36820	33402	18236	3807	71		6260	41		161566	9935	
		V_L	177603	182162	196187	216354	904		213901	937		19447	65658	
Accuracy			0.75	0.85	0.94	0.98	0.93		0.97	0.96		0.89	0.27	
Precision	Class	V_H	0.40	0.52	0.78	0.31	0.43		0.35	0.62		0.28	0.95	
		V_L	1.00	0.87	0.95	0.98	0.93		0.97	0.96		0.89	0.15	
	Average	Macro	0.70	0.70	0.87	0.64	0.68		0.66	0.79		0.59	0.55	
		Weighted	0.90	0.82	0.94	0.97	0.90		0.95	0.95		0.83	0.84	
Recall	Class	V_H	1.00	0.22	0.45	0.00	0.08		0.01	0.12		0.02	0.17	
		V_L	0.69	0.96	0.99	1.00	0.99		1.00	1.00		0.99	0.94	
	Average	Macro	0.85	0.59	0.72	0.50	0.54		0.50	0.56		0.51	0.56	
		Weighted	0.75	0.85	0.94	0.98	0.93	N / A	0.97	0.96	N / A	0.89	0.27	N / A
F1-score	Class	V_H	0.57	0.31	0.57	0.01	0.14		0.01	0.20		0.03	0.29	
		V_L	0.82	0.91	0.97	0.99	0.96		0.99	0.98		0.94	0.25	
	Average	Macro	0.77	0.64	0.79	0.56	0.60		0.57	0.66		0.55	0.55	
		Weighted	0.78	0.82	0.94	0.97	0.90		0.96	0.95		0.84	0.29	
Confusion	True Positives (TP)		123082	175488	193861	216329	896		213890	931		160723	9339	
Matrix	False Positives (FP)		104	26175	10013	3796	65		6254	36		19118	54405	
	False Negatives (FN)		54521	6674	2326	25	8		11	3		843	599	
	True Negatives (TN)		36716	7227	8223	11	6		6	5		329	11253	
False Negative Rate (FNR)			0.31	0.04	0.01	0.00	0.01		0.00	0.00		0.01	0.06	
Mean Absolute Error (MAE)			0.25	0.15	0.06	0.02	0.07		0.03	0.04		0.11	0.73	
2015-2019														
Model			y	z	aa									
Datapoints	Class	V_H	157418	21128	20364									
		V_L	156795	146961	215296									
Accuracy			0.87	0.87	0.94									
Precision	Class	V_H	0.91	0.5	0.76									
		V_L	0.83	0.9	0.95									
	Average	Macro	0.87	0.7	0.85									
		Weighted	0.87	0.85	0.93									
Recall	Class	V_H	0.81	0.3	0.4									
		V_L	0.92	0.96	0.99									
	Average	Macro	0.87	0.63	0.69					N / A				
		Weighted	0.87	0.87	0.94									
F1-score	Class	V_H	0.86	0.37	0.52									
		V_L	0.87	0.93	0.97									
	Average	Macro	0.87	0.66	0.76									
		Weighted	0.87	0.86	0.93									
Confusion	True Positives (TP)		144697	140656	212681									
Matrix	False Positives (FP)		29830	24828	12242									
	False Negatives (FN)		12098	6305	2615									
	True Negatives (TN)		127588	6300	8122									
False Negative Rate (FNR)			0.08	0.04	0.01									
Mean Absolute Error (MAE)			0.13	0.13	0.06									

[a] All outputs are defined according to Table 3.2. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] The models are trained and validated on a 10% random sample (010FD | T=12), of a a subset of the full dataset (100FD | T=12). This subset consists of all granted patents with complete fields of features, i.e. all the outputs in the time frame T=12 exist (see Fig. 4.13). This constrains the dataset to the years between 1976-2007, and we take a 10% sample from that subset to form the sample dataset (010FD | T=12). These trained and validated models are then tested on a random 20% out-of-time sample of all available output proxies from 2007-2019, in three time subsets of the full dataset (100FD) (see 4.5): (i) 2007-2011 (equivalent to Fig. 4.12 part d), (ii) 2011-2015 (equivalent to Fig. 4.12 part e), and (iii) 2015-2019 (equivalent to Fig. 4.12 part f).

[c] Class V_H represents high value patents and class V_L represents low value patents (see 3.2.2.3).

[d] The error-function derivative evaluation metrics and their definitions can be found in 4.3.

The macro average F1-score also increases marginally relative to Table 5.15, which is driven by the low precision for V_H. This indicates that these models (models d, e, f) are not able to identify clear characteristics for the V_H patent, in comparison to models p and q, where there is an increase in precision for V_H patents. These results also hold true for the generality outputs (generality_t4, generality_t8, and generality_t12), i.e. models g, h and i. We observe an improved macro average F1-score for 2011-2015 for models s and t, driven by the increase in precision and recall for the V_H class. This is partly due to the improvement in the distinction boundary between V_H and V_L patents due to improvements in digital communications, improvements in the examination process and completeness of search reports, and the introduction of the 8th edition of the IPC classification system[1] (Falk & Train, 2017).

For the renewal outputs, i.e. renewal_t4, renewal_t8, and renewal_t12, for 2007-2011, there is a reduction in the accuracy from 0.86 (model j), to 0.64 (model k), to 0.56 (model l), while the macro average F1-score remains fairly constant to 0.56. This is consistent with Table 5.15. For 2011-2015, we observe an improvement in the accuracy for renewal_t4 (model v), driven by the increase in the ratio of V_H relative to V_L patents. We also observe a significant drop in the accuracy for renewal_t8 to 0.27 (model w), driven by the very low number of V_H patents. Model w is no longer able to distinguish the characteristics of V_L and thus the number of false positives increases. This is because the trained model is tested on a non-representative testing dataset, i.e. the distribution of the output proxy in the test dataset, is not the same as the distribution of the output proxy in the training and validation datasets used to train the model.

5.2 Model out-of-sample evaluation by technological area evaluation strategy

We evaluate the forecasting ability of our models on technological areas, using a sample size evaluation strategy (see 4.5 and Fig. 4.12). We perform the out-of-sample evaluation by technological area evaluation strategy on the 10% random representative sample dataset

[1]The eighth edition of the IPC classification system came into force in 2006, where the system was revised and the classification was divided into core and advanced levels. Thus it has taken a few years for the improvement to be seen in the data.

(010FD)[1] for each IPC patent classification for all output proxies[2] (see Fig. 4.13). The purpose is to identify which output proxies are suitable for forecasting patent value for different technology areas.

5.2.1 Overall (grant_lag, generality, quality_index_4)

Table 5.2 shows the results on the IPC classification sections for the grant_lag output. All the models are able to distinguish the V_L patents from the V_H patents, shown by the consistent macro average F1-score, with the IPC G model performing well in a variety of evaluation metrics (models h_i and h_ii). The IPC section models perform worse than the full sample, due to the smaller number of datapoints. IPC G model (models h_i and h_ii) show the highest accuracy and lowest false negative rate, where as IPC D model (models e_i and e_ii) shows the lowest accuracy and the highest false negative rate. The IPC E model (models i_i and i_ii) has the highest macro average F1-score. IPC G and H models (models h_i, h_ii, i_i and i_ii) have the highest saturation in the training and validation loss comparatively to models a_i and a_ii.

Table 5.3 shows the results on the IPC classifications sections for the generality output. IPC A model (models b_i and b_ii) shows the highest accuracy, where as IPC D model shows (models e_i and e_ii) shows the lowest accuracy and highest false negative rate. However, IPC E and G models (models h_i, h_ii, i_i and i_ii) show the highest macro average F1-score with 0.69 and 0.71 respectively. IPC E model (models f_i and f_ii) also exhibits severe over fitting. This is mainly driven by the small number of datapoints and the large imbalance between V_H and V_L patents. The models are able to distinguish the V_L patents well, shown by the high F1-score for V_L patents.

Table 5.4 shows the results on the IPC classifications sections for the quality_index_4 output. The models are able to distinguish the V_L patents relative to the V_H patents, evident by the consistently high macro average F1-scores, and the high F1-score for V_L patents. They show stability and consistency for all IPC sections, driven by the composite nature of the output proxy, which makes the characteristics between V_H and V_L patents more distinct, improving the classification results (Grimaldi et al., 2018; van Zeebroeck, 2011).

[1]The 10% random representative sample dataset (010FD) is a 10% random sample of the full dataset (100FD), where year distribution and IPC distribution of patents have been stratified (Brownlee, 2017g, 2020g; Dobbin & Simon, 2011; Ng & Katanforoosh, 2020).

[2]For each output proxy, the 010FD sample dataset is split into a training dataset (98%), validation dataset (1%) and testing dataset (1%), ensuring a representative distribution of the categoric output proxy in the training, validation and testing datasets (Ng & Katanforoosh, 2020).

Table 5.2 Model evaluation on the sample dataset (010FD) for grant_lag by IPC section (technological area)

Output: Grant_Lag[a]			Full Sample		IPC Section															
					A		B		C		D		E		F		G		H	
Datapoints[b]	Class		V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L
	Total[c]		129598	389431	18113	51720	24152	58205	13366	32583	1496	2640	4192	9408	12116	27661	27700	115180	28328	91704
	Training		127019	381680	17753	50689	23671	57046	13232	32257	1466	2587	4108	9221	11875	27110	21149	112887	27764	89878
	Validation		1283	3856	179	513	239	577	134	326	15	26	42	93	120	274	274	1141	281	908
	Testing		1296	3895	181	518	242	582	135	330	15	27	42	94	121	277	277	1152	283	918
Training / Validation																				
Training Loss				0.04		18.22		13.47		33.20		102.88		81.76		39.64		3.50		5.72
Validation Loss				0.04		18.22		13.47		33.20		102.85		81.75		39.63		3.50		5.72
Training Accuracy				0.78		0.75		0.72		0.72		0.60		0.63		0.70		0.81		0.77
Validation Accuracy				0.77		0.76		0.70		0.74		0.71		0.64		0.70		0.81		0.77
Testing																				
Model			a_i	a_ii	b_i	b_ii	c_i	c_ii	d_i	d_ii	e_i	e_ii	f_i	f_ii	g_i	g_ii	h_i	h_ii	i_i	i_ii
Θ[d]			0.50	0.45	0.50	0.44	0.50	0.44	0.50	0.45	0.50	0.43	0.50	0.47	0.50	0.44	0.50	0.42	0.50	0.42
Accuracy[e]			0.77	0.76	0.75	0.73	0.72	0.69	0.72	0.70	0.60	0.62	0.72	0.74	0.69	0.68	0.82	0.76	0.77	0.74
Precision	Class	V_H	0.58	0.51	0.56	0.48	0.56	0.47	0.53	0.48	0.45	0.48	0.58	0.58	0.48	0.47	0.67	0.40	0.60	0.45
		V_L	0.79	0.84	0.77	0.82	0.73	0.79	0.75	0.79	0.73	0.79	0.75	0.82	0.73	0.80	0.82	0.86	0.78	0.83
	Average	Macro	0.68	0.68	0.67	0.65	0.65	0.63	0.64	0.64	0.59	0.64	0.67	0.70	0.61	0.64	0.75	0.63	0.69	0.64
		Weighted	0.74	0.76	0.71	0.73	0.68	0.70	0.69	0.70	0.63	0.68	0.70	0.75	0.65	0.70	0.79	0.77	0.74	0.74
Recall	Class	V_H	0.26	0.52	0.19	0.48	0.18	0.52	0.27	0.50	0.60	0.73	0.36	0.62	0.23	0.60	0.13	0.43	0.10	0.45
		V_L	0.94	0.84	0.95	0.82	0.94	0.76	0.90	0.78	0.59	0.56	0.88	0.80	0.89	0.71	0.99	0.84	0.98	0.83
	Average	Macro	0.60	0.68	0.57	0.65	0.56	0.64	0.59	0.64	0.60	0.65	0.62	0.71	0.56	0.66	0.56	0.64	0.54	0.64
		Weighted	0.77	0.76	0.75	0.73	0.72	0.69	0.72	0.70	0.60	0.62	0.72	0.74	0.69	0.68	0.82	0.76	0.77	0.74
F1-score	Class	V_H	0.36	0.51	0.28	0.48	0.27	0.49	0.36	0.49	0.51	0.58	0.44	0.60	0.31	0.53	0.22	0.41	0.17	0.45
		V_L	0.86	0.84	0.85	0.82	0.82	0.77	0.82	0.78	0.65	0.66	0.81	0.81	0.80	0.75	0.90	0.85	0.87	0.83
	Average	Macro	0.64	0.68	0.61	0.65	0.60	0.63	0.61	0.64	0.59	0.64	0.64	0.70	0.58	0.64	0.64	0.63	0.61	0.64
		Weighted	0.75	0.76	0.73	0.73	0.70	0.69	0.70	0.70	0.61	0.65	0.71	0.74	0.67	0.69	0.80	0.76	0.75	0.74
Confusion	True Positives (TP)		3643	3262	491	426	547	440	298	257	16	15	83	75	247	196	1135	971	899	761
Matrix	False Positives (FP)		954	625	147	95	198	116	99	67	6	4	27	16	93	48	242	158	255	155
	False Negatives (FN)		252	633	27	92	35	142	32	73	11	12	11	19	30	81	17	181	19	157
	True Negatives (TN)		342	671	34	86	44	126	36	68	9	11	15	26	28	73	35	119	28	128
False Negative Rate			0.06	0.16	0.05	0.18	0.06	0.24	0.10	0.22	0.41	0.44	0.12	0.20	0.11	0.29	0.01	0.16	0.02	0.17
Mean absolute error			0.23	0.24	0.25	0.27	0.28	0.31	0.28	0.30	0.40	0.38	0.28	0.26	0.31	0.32	0.18	0.24	0.23	0.26

[a] All outputs are defined according to Table 3.2. Class V_H represents high value patents and class V_L represents low value patents. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] The results are calculated on the sample dataset 010FD, which is a 10% random sample of the full dataset, 100FD, where year distribution and IPC distributions have been stratified. For every output, the number of datapoints corresponds to the total number of complete fields of features, i.e. datapoints with empty number of features because of cleaning or non-existence are not included.

[c] The sample dataset 010FD is split into three sets (maintaining the distribution of the output variable in each): training, validation and the testing set, with a ratio of 98:1:1 respectively.

[d] Θ refers to the classification threshold or decision threshold, where the model converts the probability returned from the model into a class. In the case when $\Theta = 0.50$, the model is optimised for the lowest loss function and highest accuracy. In the case, when $\Theta \neq 0.50$, the threshold is optimised to maximise the macro average F1-score (Lipton et al., 2014; Provost, 2000)

[e] All evaluation metrics of the error-function derivative and their definitions can be found in 4.3.

Table 5.3 Model evaluation on the sample dataset (010FD) for generality by IPC section (technological area)

Output: Generality[a]			Full Sample		IPC Section													
					A		B		C		D		E		F		G		H	
Datapoints[b]	Class		V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L
	Total[c]		69174	362916	6809	51773	12442	53503	8604	27436	593	2548	1408	9408	5125	26724	19251	103626	14942	87898
	Training		67797	355694	6674	50742	12194	52438	8433	26889	581	2496	1380	9219	5023	26191	18867	101564	14644	86148
	Validation		685	3593	67	513	123	530	85	272	6	26	14	94	51	265	191	1026	148	871
	Testing		692	3629	68	518	125	535	86	275	6	26	14	95	51	268	193	1036	150	879
Training / Validation																				
Training Loss				0.05		23.60		19.76		43.91		108.53		86.10		49.15		5.25		8.22
Validation Loss				0.05		23.60		19.75		43.90		108.51		86.09		49.16		5.25		8.22
Training Accuracy				0.85		0.88		0.82		0.77		0.60		0.81		0.84		0.85		0.84
Validation Accuracy				0.85		0.88		0.83		0.78		0.62		0.67		0.82		0.86		0.86
Testing																				
Model			a_i	a_ii	b_i	b_ii	c_i	c_ii	d_i	d_ii	e_i	e_ii	f_i	f_ii	g_i	g_ii	h_i	h_ii	i_i	i_ii
Θ[d]			0.50	0.41	0.50	0.44	0.50	0.45	0.50	0.45	0.50	0.48	0.50	0.49	0.50	0.45	0.50	0.43	0.50	0.41
Accuracy[e]			0.85	0.81	0.90	0.88	0.82	0.78	0.77	0.77	0.72	0.69	0.77	0.78	0.82	0.81	0.86	0.84	0.87	0.83
Precision	Class	V_H	0.61	0.44	0.64	0.46	0.54	0.43	0.54	0.51	0.36	0.33	0.32	0.34	0.38	0.40	0.59	0.50	0.67	0.42
		V_L	0.87	0.91	0.91	0.92	0.86	0.87	0.81	0.86	0.90	0.90	0.95	0.96	0.86	0.88	0.88	0.91	0.88	0.92
	Average	Macro	0.74	0.68	0.78	0.69	0.70	0.65	0.68	0.69	0.63	0.62	0.64	0.65	0.62	0.64	0.74	0.71	0.78	0.67
		Weighted	0.83	0.84	0.88	0.87	0.80	0.79	0.75	0.78	0.80	0.79	0.87	0.88	0.78	0.81	0.84	0.85	0.85	0.85
Recall	Class	V_H	0.22	0.57	0.24	0.41	0.34	0.47	0.33	0.56	0.67	0.67	0.71	0.79	0.18	0.37	0.32	0.52	0.24	0.55
		V_L	0.97	0.86	0.98	0.94	0.93	0.86	0.91	0.83	0.73	0.69	0.78	0.78	0.94	0.90	0.96	0.90	0.98	0.87
	Average	Macro	0.60	0.72	0.61	0.68	0.64	0.67	0.62	0.70	0.70	0.68	0.75	0.79	0.56	0.64	0.64	0.71	0.61	0.71
		Weighted	0.85	0.81	0.90	0.88	0.82	0.78	0.77	0.77	0.72	0.69	0.77	0.78	0.82	0.81	0.86	0.84	0.87	0.83
F1-score	Class	V_H	0.32	0.50	0.35	0.21	0.42	0.45	0.41	0.53	0.47	0.44	0.44	0.48	0.24	0.38	0.42	0.51	0.35	0.48
		V_L	0.92	0.88	0.94	0.93	0.89	0.86	0.86	0.84	0.81	0.78	0.86	0.86	0.90	0.89	0.92	0.90	0.93	0.89
	Average	Macro	0.66	0.69	0.68	0.68	0.67	0.66	0.65	0.69	0.66	0.65	0.69	0.71	0.59	0.64	0.68	0.71	0.68	0.69
		Weighted	0.84	0.82	0.89	0.87	0.81	0.78	0.76	0.77	0.76	0.74	0.82	0.83	0.80	0.81	0.84	0.84	0.86	0.84
Confusion	True Positives (TP)		3532	3126	509	485	499	458	251	229	19	18	74	74	253	240	993	933	861	767
Matrix	False Positives (FP)		540	298	52	40	83	66	58	38	2	2	4	3	42	32	131	92	114	68
	False Negatives (FN)		97	503	9	33	36	77	24	46	7	8	21	21	15	28	43	103	18	112
	True Negatives (TN)		152	394	16	28	42	55	28	48	4	4	10	11	9	19	62	101	36	82
False Negative Rate (FNR)			0.03	0.14	0.02	0.06	0.07	0.14	0.09	0.17	0.27	0.31	0.22	0.22	0.06	0.10	0.04	0.10	0.02	0.13
Mean Absolute Error (MAE)			0.06	0.19	0.10	0.12	0.18	0.22	0.23	0.23	0.28	0.31	0.23	0.22	0.18	0.19	0.14	0.16	0.13	0.17

[a] All outputs are defined according to Table 3.2. Class V_H represents high value patents and class V_L represents low value patents. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] The results are calculated on the sample dataset 010FD, which is a 10% random sample of the full dataset, 100FD, where year distribution and IPC distributions have been stratified. For every output, the number of datapoints corresponds to the total number of complete fields of features, i.e. datapoints with empty number of features because of cleaning or non-existence are not included.

[c] The sample dataset 010FD is split into three sets (maintaining the distribution of the output variable in each): training, validation and the testing set, with a ratio of 98:1:1 respectively.

[d] Θ refers to the classification threshold or decision threshold, where the model converts the probability returned from the model into a class. In the case when $\Theta = 0.50$, the model is optimised for the lowest loss function and highest accuracy. In the case, when $\Theta \neq 0.50$, the threshold is optimised to maximise the macro average F1-score (Lipton et al., 2014; Provost, 2000).

[e] All evaluation metrics of the error-function derivative and their definitions can be found in 4.3.

Table 5.4 Model evaluation on the sample dataset (010FD) for quality_index_4 by IPC section (technological area)

Output: Quality_Index_4[a]			Full Sample		IPC Section													
					A		B		C		D		E		F		G		H	
Datapoints[b]	Class		V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L
	Total[c]		42658	476376	5622	64211	7288	75070	4793	41621	373	3763	884	12716	3362	36416	10606	132277	9730	110302
	Training		41809	466895	5510	62932	7143	73575	4697	40792	365	3688	866	12463	3295	35691	10395	129644	9537	108105
	Validation		422	4717	56	636	72	744	48	412	4	37	9	126	33	361	105	1310	96	1093
	Testing		427	4764	56	643	73	751	48	417	4	38	9	127	34	364	106	1323	97	1104
Training / Validation																				
Training Loss				0.02		17.74		13.07		32.67		102.54		80.42		38.70		3.39		5.52
Validation Loss				0.02		17.74		13.08		32.67		102.57		80.44		38.70		3.39		5.53
Training Accuracy				0.94		0.94		0.95		0.93		0.68		0.93		0.95		0.95		0.95
Validation Accuracy				0.94		0.94		0.93		0.91		0.58		0.74		0.92		0.95		0.95
Testing																				
Model					a_i	a_ii	b_i	b_ii	c_i	c_ii	d_i	d_ii	e_i	e_ii	f_i	f_ii	g_i	g_ii	h_i	h_ii	i_i	i_ii
Θ[d]					0.50	0.38	0.50	0.43	0.50	0.45	0.50	0.45	0.50	0.47	0.50	0.49	0.50	0.43	0.50	0.45	0.50	0.43
Accuracy[e]					0.95	0.93	0.95	0.95	0.93	0.93	0.92	0.92	0.74	0.71	0.93	0.93	0.91	0.91	0.95	0.95	0.95	0.94
Precision	Class	V_H			0.80	0.56	0.79	0.65	0.71	0.63	0.70	0.63	0.18	0.21	0.60	0.50	0.50	0.47	0.76	0.70	0.80	0.66
		V_L			0.95	0.97	0.96	0.97	0.94	0.95	0.93	0.95	0.94	0.96	0.98	0.98	0.93	0.96	0.96	0.97	0.95	0.97
	Average	Macro			0.88	0.77	0.88	0.81	0.83	0.79	0.82	0.79	0.56	0.59	0.79	0.74	0.72	0.72	0.86	0.84	0.88	0.82
		Weighted			0.94	0.94	0.94	0.95	0.92	0.93	0.91	0.92	0.86	0.89	0.95	0.95	0.90	0.92	0.95	0.95	0.94	0.94
Recall	Class	V_H			0.46	0.73	0.48	0.71	0.37	0.52	0.40	0.60	0.50	0.75	0.67	0.78	0.26	0.59	0.49	0.60	0.44	0.65
		V_L			0.99	0.95	0.99	0.97	0.99	0.97	0.98	0.96	0.76	0.71	0.97	0.94	0.98	0.94	0.99	0.98	0.99	0.97
	Average	Macro			0.73	0.84	0.74	0.84	0.68	0.75	0.69	0.78	0.63	0.73	0.82	0.86	0.62	0.77	0.74	0.79	0.72	0.81
		Weighted			0.95	0.93	0.95	0.95	0.93	0.93	0.92	0.92	0.74	0.71	0.95	0.93	0.91	0.91	0.95	0.95	0.95	0.94
F1-score	Class	V_H			0.58	0.63	0.60	0.68	0.49	0.57	0.51	0.61	0.26	0.33	0.63	0.61	0.34	0.52	0.60	0.65	0.57	0.65
		V_L			0.97	0.96	0.97	0.97	0.96	0.96	0.95	0.95	0.84	0.82	0.97	0.96	0.95	0.95	0.97	0.97	0.97	0.97
	Average	Macro			0.79	0.80	0.80	0.82	0.75	0.77	0.75	0.78	0.59	0.65	0.80	0.80	0.66	0.74	0.80	0.81	0.79	0.81
		Weighted			0.94	0.93	0.94	0.95	0.92	0.93	0.91	0.92	0.80	0.79	0.95	0.94	0.90	0.91	0.95	0.95	0.94	0.94
Confusion	True Positives (TP)				4714	4524	636	621	740	729	409	400	29	27	120	120	355	341	1307	1295	1093	1071
Matrix	False Positives (FP)				230	116	29	16	46	35	29	19	2	1	2	2	25	14	54	42	54	34
	False Negatives (FN)				50	240	7	22	11	22	8	17	9	11	7	7	9	23	16	28	11	33
	True Negatives (TN)				197	311	27	40	27	38	19	29	2	3	7	7	9	20	52	64	43	63
False Negative Rate (FNR)					0.01	0.05	0.01	0.03	0.01	0.03	0.02	0.04	0.24	0.29	0.06	0.06	0.02	0.06	0.01	0.02	0.01	0.03
Mean Absolute Error (MAE)					0.05	0.07	0.05	0.05	0.07	0.07	0.08	0.08	0.26	0.29	0.07	0.07	0.09	0.09	0.05	0.05	0.05	0.06

[a] All outputs are defined according to Table 3.2. Class V_H represents high value patents and class V_L represents low value patents. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] The results are calculated on the sample dataset 010FD, which is a 10% random sample of the full dataset, 100FD, where year distribution and IPC distributions have been stratified. For every output, the number of datapoints corresponds to the total number of complete fields of features, i.e. datapoints with empty number of features because of cleaning or non-existence are not included.

[c] The sample dataset 010FD is split into three sets (maintaining the distribution of the output variable in each): training, validation and the testing set, with a ratio of 98:1:1 respectively.

[d] Θ refers to the classification threshold or decision threshold, where the model converts the probability returned from the model into a class. In the case when $\Theta = 0.50$, the model is optimised for the lowest loss function and highest accuracy. In the case, when $\Theta \neq 0.50$, the threshold is optimised to maximise the macro average F1-score (Lipton et al., 2014; Provost, 2000).

[e] All evaluation metrics of the error-function derivative and their definitions can be found in 4.3.

5.2.2 Forward citations (citations_t4, citations_t8, citations_t12)

Table 5.5 shows the results on the IPC classification sections for the citation_t4 output. The models performed worse when split into individual IPC section models, rather than when combined all together. We observe a drop in the macro average F1-score, and the relative increase in the mean absolute error. We also observe the following: IPC G and H models (models h_i, h_ii, i_i, and i_ii), have the highest share of V_H patents, and the lowest training and validation losses, indicating that the convergence of weights is more robust. In addition, the IPC A model (models b_i and b_ii) follows IPC G and H, with the highest share of V_H patents, yet the training and validation loss is higher than that of IPC B model (models c_i and c_ii), indicating that these patents are more difficult to be distinguished.

Table 5.6 shows the results on the IPC classification sections for the citation_t8 output. These follow a similar pattern as before when the separation in the individual IPC section models occur. However, the range of the results for the training and validation is smaller than that of the results for citations_t4. While we observe an increase in the mean absolute error and false negative rate in all IPC sections relative to Table 5.5, and a slight expectable decrease in the overall accuracy, we can see an increase in the macro average F1-score with the highest observed in IPC A (models b_i and b_ii). This is mainly driven by the high precision for V_H, which is the result of the increase of V_H patents. This leads to a more distinct separation between V_H and V_L patents.

Table 5.7 shows the results on the IPC classification sections for the citations_t12 output. We observe that the model on the full sample has a high accuracy of 0.88 and a macro average F1-score of 0.70. IPC B, C, and F models (models c_i, c_ii, d_i, d_ii, g_i and g_ii) have a higher accuracy than the full sample model, but the macro average F1-score is lower. This driven by the low precision and recall for V_H patents. IPC A, G, and H models (models b_i, b_ii, h_i, h_ii, i_i, and i_ii) have a relative lower accuracy than the full sample accuracy, but a macro average F1-score higher, which is driven by the higher number of available V_H patents. The false negative rate is higher than that of citations_t8, which is expected as the forecasting ability of the model in T=12 decreases.

Table 5.5 Model evaluation on the sample dataset (010FD) for citations_t4 by IPC section (technological area)

Output: Citation_t4[a]			Full Sample		IPC Section													
					A		B		C		D		E		F		G		H	
Datapoints[b]	Class		V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L
	Total[c]		4015	415426	783	55452	195	70236	132	39499	9	3748	96	11081	110	32556	1625	111137	1065	91717
	Training		3935	407158	767	54348	191	68837	130	38711	9	3672	94	10860	108	31907	1593	108924	1043	89892
	Validation		40	4113	8	549	2	696	1	392	0	38	1	110	1	323	16	1101	11	908
	Testing		40	4155	8	555	2	703	1	397	0	38	1	111	1	326	16	1112	11	917
Training / Validation																				
Training Loss				0.04		24.90		17.45		38.95		103.17		84.40		46.88		6.45		10.10
Validation Loss				0.04		24.90		17.45		38.95		103.14		84.39		46.88		6.45		10.10
Training Accuracy				0.99		0.99		1.00		0.99		0.53		0.70		0.99		0.99		0.99
Validation Accuracy				0.99		0.99		1.00		1.00		0.71		0.95		1.00		0.98		0.99
Testing																				
Model			a_i	a_ii	b_i	b_ii	c_i	c_ii	d_i	d_ii	e_i	e_ii	f_i	f_ii	g_i	g_ii	h_i	h_ii	i_i	i_ii
Θ[d]			0.50	0.27	0.50	0.36	0.50	0.33	0.50	0.40	0.50	0.49	0.50	0.49	0.50	0.41	0.50	0.29	0.50	0.39
Accuracy[e]			0.99	0.97	0.99	0.96	1.00	1.00	1.00	1.00	0.58	0.50	0.94	0.92	1.00	1.00	0.99	0.92	0.99	0.99
Precision	Class	V_H	0.01	0.14	0.01	0.16	0.01	0.50	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.07	0.01	1.00
		V_L	0.99	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00	1.00	0.99	0.99	0.99	0.99
	Average	Macro	0.50	0.57	0.50	0.57	0.50	0.75	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.49	0.53	0.49	0.99
		Weighted	0.98	0.99	0.97	0.98	0.99	1.00	0.99	0.99	1.00	1.00	0.98	0.98	0.99	0.99	0.97	0.98	0.98	0.99
Recall	Class	V_H	0.01	0.45	0.01	0.38	0.01	0.50	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.38	0.01	0.09
		V_L	1.00	0.97	1.00	0.97	1.00	1.00	1.00	1.00	0.58	0.50	0.95	0.93	1.00	1.00	1.00	0.93	1.00	1.00
	Average	Macro	0.50	0.71	0.51	0.68	0.51	0.75	0.51	0.51	0.30	0.26	0.48	0.47	0.51	0.51	0.51	0.66	0.51	0.55
		Weighted	0.99	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
F1-score	Class	V_H	0.01	0.21	0.01	0.23	0.01	0.50	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.12	0.01	0.17
		V_L	0.99	0.98	0.99	0.98	1.00	1.00	1.00	1.00	0.73	0.67	0.97	0.96	1.00	1.00	0.99	0.96	0.99	0.99
	Average	Macro	0.50	0.63	0.50	0.62	0.50	0.75	0.50	0.50	0.37	0.33	0.48	0.48	0.50	0.50	0.50	0.54	0.50	0.58
		Weighted	0.98	0.98	0.97	0.97	1.00	1.00	1.00	1.00	0.73	0.67	0.96	0.95	1.00	1.00	0.98	0.95	0.98	0.98
Confusion	True Positives (TP)		4153	4047	555	539	703	702	396	396	22	19	105	103	326	326	1112	1034	917	917
Matrix	False Positives (FP)		38	22	7	5	2	1	1	1	0	0	1	1	1	1	16	10	11	10
	False Negatives (FN)		2	108	0	16	0	1	0	0	16	19	6	8	0	0	0	78	0	0
	True Negatives (TN)		2	18	1	3	0	1	0	1	0	0	0	0	0	0	0	6	0	1
False Negative Rate (FNR)			0.00	0.03	0.00	0.03	0.00	0.00	0.00	0.00	0.42	0.50	0.05	0.07	0.00	0.00	0.00	0.07	0.00	0.00
Mean Absolute Error (MAE)			0.01	0.03	0.01	0.04	0.00	0.00	0.00	0.00	0.42	0.50	0.06	0.08	0.00	0.00	0.01	0.08	0.01	0.01

[a] All outputs are defined according to Table 3.2. Class V_H represents high value patents and class V_L represents low value patents. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] The results are calculated on the sample dataset 010FD, which is a 10% random sample of the full dataset, 100FD, where year distribution and IPC distributions have been stratified. For every output, the number of datapoints corresponds to the total number of complete fields of features, i.e. datapoints with empty number of features because of cleaning or non-existence are not included.

[c] The sample dataset 010FD is split into three sets (maintaining the distribution of the output variable in each): training, validation and the testing set, with a ratio of 98:1:1 respectively.

[d] Θ refers to the classification threshold or decision threshold, where the model converts the probability returned from the model into a class. In the case when $\Theta = 0.50$, the model is optimised for the lowest loss function and highest accuracy. In the case, when $\Theta \neq 0.50$, the threshold is optimised to maximise the macro average F1-score (Lipton et al., 2014; Provost, 2000)

[e] All evaluation metrics of the error-function derivative and their definitions can be found in 4.3.

Table 5.6 Model evaluation on the sample dataset (010FD) for citations_t8 by IPC section (technological area)

Output: Citation_t8[a]			Full Sample		IPC Section													
					A		B		C		D		E		F		G		H	
Datapoints[b]	Class		V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L
	Total[c]		21202	301497	3741	39293	1652	56950	929	32385	70	3261	385	8690	639	26039	7915	73040	5871	61575
	Training		20780	295497	3667	38509	1619	55815	911	31739	68	3196	377	8517	627	25778	7758	71585	5754	60349
	Validation		210	2985	37	390	16	565	9	321	1	32	4	86	6	261	78	724	58	610
	Testing		212	3015	37	394	17	570	9	325	1	33	4	87	6	264	79	731	59	616
Training / Validation																				
Training Loss				0.15		35.53		23.32		46.30		103.92		89.58		55.10		13.61		18.79
Validation Loss				0.15		35.53		23.33		46.30		103.89		89.59		55.09		13.61		18.79
Training Accuracy				0.94		0.92		0.97		0.96		0.55		0.65		0.94		0.91		0.91
Validation Accuracy				0.94		0.92		0.97		0.98		0.64		0.90		0.98		0.90		0.91
Testing																				
Model			a_i	a_ii	b_i	b_ii	c_i	c_ii	d_i	d_ii	e_i	e_ii	f_i	f_ii	g_i	g_ii	h_i	h_ii	i_i	i_ii
Θ[d]			0.50	0.37	0.50	0.47	0.50	0.43	0.50	0.41	0.50	0.49	0.50	0.46	0.50	0.42	0.50	0.40	0.50	0.41
Accuracy[e]			0.94	0.90	0.94	0.94	0.97	0.97	0.97	0.95	0.74	0.71	0.86	0.73	0.97	0.96	0.91	0.89	0.92	0.91
Precision	Class	V_H	0.59	0.34	0.73	0.66	1.00	0.62	0.01	0.18	0.10	0.09	0.01	0.08	0.01	0.20	0.68	0.42	0.80	0.48
		V_L	0.94	0.96	0.95	0.96	0.97	0.98	0.97	0.98	1.00	1.00	0.95	0.97	0.98	0.98	0.92	0.94	0.92	0.94
	Average	Macro	0.77	0.65	0.84	0.81	0.99	0.80	0.49	0.58	0.55	0.55	0.48	0.52	0.49	0.59	0.80	0.68	0.86	0.71
		Weighted	0.92	0.92	0.93	0.93	0.97	0.97	0.95	0.96	0.97	0.97	0.91	0.93	0.96	0.97	0.90	0.89	0.91	0.90
Recall	Class	V_H	0.10	0.49	0.43	0.57	0.06	0.29	0.01	0.22	1.00	1.00	0.01	0.50	0.01	0.33	0.22	0.48	0.07	0.37
		V_L	1.00	0.93	0.98	0.97	1.00	0.99	1.00	0.97	0.73	0.70	0.90	0.74	1.00	0.97	0.99	0.93	1.00	0.96
	Average	Macro	0.55	0.71	0.71	0.77	0.53	0.64	0.50	0.60	0.86	0.85	0.45	0.62	0.50	0.65	0.60	0.70	0.53	0.67
		Weighted	0.94	0.90	0.94	0.94	0.97	0.97	0.97	0.95	0.74	0.71	0.86	0.73	0.97	0.96	0.91	0.89	0.92	0.91
F1-score	Class	V_H	0.17	0.40	0.54	0.61	0.11	0.40	0.01	0.20	0.18	0.17	0.01	0.14	0.01	0.25	0.33	0.45	0.13	0.42
		V_L	0.97	0.94	0.96	0.96	0.98	0.98	0.98	0.97	0.84	0.82	0.92	0.84	0.99	0.97	0.95	0.93	0.96	0.95
	Average	Macro	0.64	0.68	0.77	0.79	0.69	0.71	0.49	0.59	0.67	0.67	0.46	0.57	0.49	0.62	0.69	0.69	0.66	0.69
		Weighted	0.93	0.91	0.93	0.93	0.97	0.97	0.96	0.95	0.84	0.82	0.88	0.82	0.96	0.96	0.90	0.89	0.91	0.90
Confusion	True Positives (TP)		3643	2812	388	383	570	567	325	316	24	23	78	64	263	256	723	679	615	592
Matrix	False Positives (FP)		954	108	21	16	16	12	9	7	0	0	4	2	6	4	62	41	55	37
	False Negatives (FN)		252	203	6	11	0	3	0	9	9	10	9	23	1	8	8	52	1	24
	True Negatives (TN)		342	104	16	21	1	5	0	2	1	1	0	2	0	2	17	38	4	22
False Negative Rate (FNR)			0.06	0.07	0.02	0.03	0.00	0.01	0.00	0.03	0.27	0.30	0.10	0.26	0.00	0.03	0.01	0.07	0.00	0.04
Mean Absolute Error (MAE)			0.06	0.10	0.06	0.06	0.03	0.03	0.03	0.05	0.26	0.29	0.14	0.27	0.03	0.04	0.09	0.11	0.08	0.09

[a] All outputs are defined according to Table 3.2. Class V_H represents high value patents and class V_L represents low value patents. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] The results are calculated on the sample dataset 010FD, which is a 10% random sample of the full dataset, 100FD, where year distribution and IPC distributions have been stratified. For every output, the number of datapoints corresponds to the total number of complete fields of features, i.e. datapoints with empty number of features because of cleaning or non-existence are not included.

[c] The sample dataset 010FD is split into three sets (maintaining the distribution of the output variable in each): training, validation and the testing set, with a ratio of 98:1:1 respectively.

[d] Θ refers to the classification threshold or decision threshold, where the model converts the probability returned from the model into a class. In the case when $\Theta = 0.50$, the model is optimised for the lowest loss function and highest accuracy. In the case, when $\Theta \neq 0.50$, the threshold is optimised to maximise the macro average F1-score (Lipton et al., 2014; Provost, 2000).

[e] All evaluation metrics of the error-function derivative and their definitions can be found in 4.3.

Table 5.7 Model evaluation on the sample dataset (010FD) for citations_t12 by IPC section (technological area)

Output: Citation_t12[a]			Full Sample		IPC Section													
					A		B		C		D		E		F		G		H	
Datapoints[b]	Class		V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L
	Total[c]		34089	221984	6408	29189	3727	46283	1942	26689	138	2817	699	6975	1268	21774	11136	47403	8771	40854
	Training		33410	217566	6280	28608	3653	45360	1904	26156	136	2759	685	6836	1242	21340	10915	46458	8596	40040
	Validation		338	2198	64	289	37	459	19	265	1	29	7	69	13	216	110	470	87	405
	Testing		341	2220	64	292	37	459	19	268	1	29	7	70	13	216	111	475	88	409
Training / Validation																				
Training Loss				0.42		43.84		29.61		52.88		105.23		93.35		61.22		24.00		30.36
Validation Loss				0.42		43.83		29.60		52.87		105.20		93.34		61.21		24.00		30.36
Training Accuracy				0.88		0.84		0.93		0.90		0.55		0.63		0.88		0.83		0.83
Validation Accuracy				0.88		0.86		0.92		0.93		0.77		0.61		0.95		0.83		0.85
Testing																				
Model			a_i	a_ii	b_i	b_ii	c_i	c_ii	d_i	d_ii	e_i	e_ii	f_i	f_ii	g_i	g_ii	h_i	h_ii	i_i	i_ii
Θ[d]			0.50	0.42	0.50	0.44	0.50	0.39	0.50	0.44	0.50	0.49	0.50	0.49	0.50	0.46	0.50	0.44	0.50	0.44
Accuracy[e]			0.88	0.86	0.86	0.85	0.92	0.87	0.93	0.90	0.63	0.63	0.81	0.81	0.92	0.91	0.84	0.82	0.84	0.83
Precision	Class	V_H	0.64	0.47	0.65	0.56	0.01	0.26	0.25	0.29	0.08	0.08	0.28	0.28	0.01	0.23	0.67	0.54	0.61	0.53
		V_L	0.89	0.91	0.90	0.93	0.93	0.95	0.94	0.95	1.00	1.00	0.97	0.97	0.94	0.95	0.86	0.88	0.87	0.88
	Average	Macro	0.78	0.69	0.77	0.74	0.46	0.61	0.59	0.62	0.54	0.54	0.62	0.62	0.47	0.59	0.76	0.71	0.74	0.70
		Weighted	0.86	0.85	0.85	0.86	0.86	0.90	0.94	0.91	0.97	0.97	0.90	0.90	0.89	0.91	0.82	0.81	0.82	0.82
Recall	Class	V_H	0.24	0.43	0.52	0.69	0.01	0.41	0.05	0.32	1.00	1.00	0.71	0.71	0.01	0.23	0.32	0.45	0.32	0.42
		V_L	0.98	0.93	0.94	0.88	1.00	0.91	0.90	0.94	0.62	0.62	0.81	0.81	0.98	0.95	0.96	0.91	0.96	0.92
	Average	Macro	0.63	0.68	0.73	0.78	0.50	0.66	0.52	0.63	0.81	0.81	0.76	0.76	0.49	0.59	0.64	0.68	0.64	0.67
		Weighted	0.88	0.86	0.86	0.85	0.92	0.87	0.93	0.90	0.63	0.63	0.81	0.81	0.92	0.91	0.84	0.82	0.84	0.83
F1-score	Class	V_H	0.35	0.45	0.58	0.62	0.01	0.32	0.08	0.30	0.15	0.15	0.40	0.40	0.01	0.23	0.43	0.49	0.42	0.47
		V_L	0.93	0.92	0.92	0.90	0.96	0.93	0.92	0.94	0.77	0.77	0.88	0.88	0.96	0.95	0.91	0.89	0.91	0.90
	Average	Macro	0.70	0.68	0.75	0.76	0.48	0.63	0.55	0.62	0.65	0.65	0.68	0.68	0.48	0.59	0.69	0.69	0.69	0.68
		Weighted	0.87	0.85	0.85	0.85	0.89	0.88	0.93	0.90	0.76	0.76	0.85	0.85	0.90	0.91	0.83	0.81	0.83	0.82
Confusion	True Positives (TP)		2174	2057	274	257	463	422	265	253	18	18	57	57	213	208	457	432	391	376
Matrix	False Positives (FP)		260	196	31	20	37	22	18	13	0	0	2	2	13	10	75	61	60	51
	False Negatives (FN)		46	163	18	35	1	42	3	15	11	11	13	13	5	10	18	43	18	33
	True Negatives (TN)		81	145	33	44	0	15	1	6	1	1	5	5	0	3	36	50	28	37
False Negative Rate			0.02	0.07	0.06	0.12	0.00	0.09	0.01	0.06	0.38	0.38	0.19	0.19	0.02	0.05	0.04	0.09	0.04	0.08
Mean absolute error			0.12	0.14	0.14	0.15	0.08	0.13	0.07	0.10	0.37	0.37	0.19	0.19	0.08	0.09	0.16	0.18	0.16	0.17

[a] All outputs are defined according to Table 3.2. Class V_H represents high value patents and class V_L represents low value patents. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] The results are calculated on the sample dataset 010FD, which is a 10% random sample of the full dataset, 100FD, where year distribution and IPC distributions have been stratified. For every output, the number of datapoints corresponds to the total number of complete fields of features, i.e. datapoints with empty number of features because of cleaning or non-existence are not included.

[c] The sample dataset 010FD is split into three sets (maintaining the distribution of the output variable in each): training, validation and the testing set, with a ratio of 98:1:1 respectively.

[d] Θ refers to the classification threshold or decision threshold, where the model converts the probability returned from the model into a class. In the case when $\Theta = 0.50$, the model is optimised for the lowest loss function and highest accuracy. In the case, when $\Theta \neq 0.50$, the threshold is optimised to maximise the macro average F1-score (Lipton et al., 2014; Provost, 2000)

[e] All evaluation metrics of the error-function derivative and their definitions can be found in 4.3

5.2.3 Generality index (generality_t4, generality_t8, generality_t12)

Table 5.8 shows the results on the IPC sections for the generality_t4 output. The individual IPC models perform similarly to the full combined sample. We also observe the following: while the IPC G and H models (models h_i, h_ii, i_i and i_ii) have the lowest training and validation loss, indicating that the convergence of weights is more robust, the IPC C model (models d_i and d_ii) has the optimal performance for both accuracy and macro average F1-score. This indicates that the distinction boundary between V_H and V_L patents is clearer for IPC section C.

Table 5.9 shows the results on the IPC sections for the generality_t8 output, which are similar to Table 5.8. The individual IPC section models perform similarly to the full combined sample. IPC A model (models b_i and b_ii) has the optimal performance for both accuracy and macro F1-score. This indicates that the distinction boundary between V_H and V_L patents is clearer for IPC A, rather than for example IPC B, G and H, (models c_i, c_ii, h_i, h_ii, i_i and i_ii), which have the highest number of V_H patents. In addition, the IPC D, E, and F models (models e_i, e_ii, f_i, f_ii, g_i, and g_ii) are slightly overfitted.

Table 5.10 shows the results on the IPC sections for the generality_t12 output. IPC G model (models h_i and h_ii) has the optimal performance for both accuracy and macro average F1-score. This indicates that the distinction boundary between V_H and V_L patents is clearer for IPC G, rather than for example IPC section B, which has the highest number of V_H patents. The models for IPC D and E (models e_i, e_ii, f_i and f_ii) have the highest false negative rate, driven by the higher number of false negatives and the low number of datapoints. The IPC D model (models e_i and e_ii) is slightly overfitted, where as the IPC E model (models f_i and f_ii) is slightly underfitted.

5.2.4 Renewals (renewal_t4, renewal_t8, renewal_t12)

Table 5.11 shows the results on the IPC sections for the renewal_t4 output. The individual IPC section models perform similarly to the full combined sample. While IPC G model (models h_i and h_ii), has the lowest training and validation loss, IPC H model (models i_i and i_ii) has the optimal performance for both accuracy and macro average F1-score, despite not having the highest number of V_H patents. This indicates that the distinction boundary between V_H and V_L patents is clearer for IPC H. Table 5.12 shows the results for the renewal_t8 output. The results are similar to Table 5.11, with IPC G model (models h_i and h_ii), having the optimal performance for macro average F1-score optimisation. This is also similar to Table 5.13 for the renewal_t12 output. The false negative rate rises further, driven by the number of false negatives and the low recall values.

Table 5.8 Model evaluation on the sample dataset 010FD for generality_t4 by IPC section (technological area)

Output: Generality_t4[a]			Full Sample		IPC Section													
					A		B		C		D		E		F		G		H	
Datapoints[b]	Class[c]		V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L
	Total[d]		9612	409829	2203	54032	2122	68309	329	39302	50	3707	407	10770	745	31921	2045	110717	1711	91071
	Training		9421	401672	2159	52956	2080	66948	323	38518	48	3633	399	10555	731	31284	2005	108512	1677	89258
	Validation		95	4058	22	535	21	677	3	390	1	37	1	107	7	317	20	1097	17	902
	Testing		96	4099	22	541	21	684	3	394	1	37	1	108	7	320	20	1108	17	911
Training / Validation																				
Training Loss				0.04		25.06		17.43		39.02		102.92		84.68		46.90		6.45		10.14
Validation Loss				0.04		25.06		17.43		39.03		102.89		84.68		36.90		6.45		10.14
Training Accuracy				0.97		0.96		0.96		0.99		0.56		0.70		0.98		0.98		0.98
Validation Accuracy				0.97		0.96		0.96		0.99		0.57		0.94		0.98		0.98		0.98
Testing																				
Model			a_i	a_ii	b_i	b_ii	c_i	c_ii	d_i	d_ii	e_i	e_ii	f_i	f_ii	g_i	g_ii	h_i	h_ii	i_i	i_ii
Θ[e]			0.50	0.35	0.50	0.39	0.50	0.36	0.50	0.34	0.50	0.49	0.50	0.47	0.50	0.45	0.50	0.37	0.50	0.31
Accuracy[f]			0.98	0.97	0.96	0.94	0.97	0.95	0.99	0.96	0.68	0.61	0.89	0.82	0.98	0.98	0.98	0.98	0.98	0.92
Precision	Class	V_H	0.01	0.34	0.01	0.24	0.01	0.18	0.01	0.23	0.08	0.06	0.01	0.10	0.01	0.01	0.01	0.40	0.01	0.09
		V_L	0.98	0.98	0.96	0.97	0.97	0.98	0.99	0.98	1.00	1.00	0.96	0.98	0.98	0.98	0.98	0.99	0.98	0.99
	Average	Macro	0.50	0.66	0.49	0.60	0.49	0.58	0.50	0.61	0.54	0.53	0.48	0.54	0.49	0.49	0.49	0.69	0.49	0.54
		Weighted	0.95	0.97	0.92	0.94	0.94	0.95	0.98	0.97	0.98	0.98	0.93	0.95	0.96	0.96	0.96	0.98	0.96	0.97
Recall	Class	V_H	0.01	0.17	0.01	0.18	0.01	0.19	0.01	0.31	1.00	1.00	0.01	0.50	0.01	0.01	0.01	0.20	0.01	0.35
		V_L	1.00	0.99	1.00	0.98	1.00	0.97	1.00	0.97	0.68	0.59	0.93	0.83	1.00	1.00	1.00	0.99	1.00	0.93
	Average	Macro	0.50	0.58	0.50	0.58	0.50	0.58	0.50	0.64	0.84	0.80	0.46	0.67	0.50	0.50	0.50	0.60	0.50	0.64
		Weighted	0.98	0.97	0.96	0.94	0.97	0.95	0.99	0.96	0.68	0.61	0.89	0.82	0.98	0.98	0.98	0.98	0.98	0.92
F1-score	Class	V_H	0.35	0.23	0.01	0.21	0.01	0.18	0.01	0.26	0.15	0.11	0.01	0.17	0.01	0.01	0.01	0.27	0.01	0.14
		V_L	0.93	0.98	0.98	0.97	0.98	0.97	0.99	0.97	0.81	0.74	0.94	0.90	0.99	0.99	0.99	0.99	0.99	0.96
	Average	Macro	0.50	0.62	0.49	0.59	0.49	0.58	0.50	0.62	0.66	0.64	0.47	0.60	0.49	0.49	0.49	0.64	0.49	0.59
		Weighted	0.96	0.97	0.94	0.94	0.95	0.95	0.98	0.96	0.80	0.75	0.91	0.88	0.97	0.97	0.97	0.98	0.97	0.94
Confusion	True Positives (TP)		4099	4068	541	528	684	666	394	390	25	22	100	90	320	320	1108	1102	911	850
Matrix	False Positives (FP)		95	80	22	18	21	17	3	3	0	0	4	2	7	7	20	16	17	11
	False Negatives (FN)		0	31	0	13	0	18	0	4	12	15	8	18	0	0	0	6	0	61
	True Negatives (TN)		1	16	0	4	0	4	0	0	1	1	0	2	0	0	0	1102	0	6
False Negative Rate (FNR)			0.00	0.01	0.00	0.02	0.00	0.03	0.00	0.01	0.32	0.41	0.07	0.17	0.00	0.00	0.00	0.01	0.00	0.07
Mean Absolute Error (MAE)			0.02	0.03	0.04	0.06	0.03	0.05	0.01	0.04	0.32	0.39	0.11	0.18	0.02	0.02	0.02	0.02	0.02	0.08

[a] All outputs are defined according to Table 3.2. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] The results are calculated on the sample dataset 010FD, which is a 10% random sample of the full dataset, 100FD, where year distribution and IPC distributions have been stratified. For every output, the number of datapoints corresponds to the total number of complete fields of features, i.e. datapoints with empty number of features because of cleaning or non-existence are not included.

[c] Class V_H represents high value patents and class V_L represents low value patents.

[d] The sample dataset 010FD is split into three sets (maintaining the distribution of the output variable in each): training, validation and the testing set, with a ratio of 98:1:1 respectively.

[e] Θ refers to the classification threshold or decision threshold, where the model converts the probability returned from the model into a class. In the case when $\Theta = 0.50$, the model is optimised for the lowest loss function and highest accuracy. In the case, when $\Theta \neq 0.50$, the threshold is optimised to maximise the macro average F1-score (Lipton et al., 2014; Provost, 2000).

[f] All evaluation metrics of the error-function derivative and their definitions can be found in 4.3.

Table 5.9 Model evaluation on the sample dataset 010FD for generality_t8 by IPC section (technological area)

Output: Generality_t8[a]			Full Sample		IPC Section													
					A		B		C		D		E		F		G		H	
Datapoints[b]	Class[c]		V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L
	Total[d]		18906	303793	2895	40139	4184	53873	1334	32525	137	3194	660	8415	1594	25348	4223	76732	3879	63567
	Training		18530	297747	2837	39339	4101	53333	1308	31350	135	3129	646	8248	1562	24843	4139	75204	3802	62301
	Validation		187	3008	29	398	41	540	13	317	1	32	7	83	16	251	42	760	38	630
	Testing		189	3038	29	402	42	545	13	321	1	33	7	84	16	254	42	768	39	636
Training / Validation																				
Training Loss				0.15		35.53		23.46		46.43		103.86		89.74		55.39		13.53		18.72
Validation Loss				0.15		35.52		23.46		46.43		103.82		.89.73		55.39		13.53		18.72
Training Accuracy				0.94		0.93		0.93		0.95		0.56		0.64		0.90		0.95		0.94
Validation Accuracy				0.94		0.93		0.93		0.96		0.70		0.74		0.94		0.95		0.94
Testing																				
Model			a_i	a_ii	b_i	b_ii	c_i	c_ii	d_i	d_ii	e_i	e_ii	f_i	f_ii	g_i	g_ii	h_i	h_ii	i_i	i_ii
Θ[e]			0.50	0.36	0.50	0.39	0.50	0.40	0.50	0.40	0.50	0.49	0.50	0.46	0.50	0.42	0.50	0.35	0.50	0.38
Accuracy[f]			0.94	0.90	0.94	0.89	0.93	0.93	0.96	0.91	0.56	0.56	0.80	0.67	0.94	0.86	0.95	0.88	0.94	0.90
Precision	Class	V_H	0.17	0.23	1.00	0.27	0.01	0.47	0.01	0.16	0.06	0.06	0.13	0.17	0.50	0.19	0.01	0.22	0.50	0.27
		V_L	0.94	0.96	0.94	0.97	0.93	0.95	0.96	0.97	1.00	1.00	0.93	0.98	0.95	0.96	0.95	0.97	0.94	0.96
	Average	Macro	0.55	0.60	0.97	0.62	0.47	0.71	0.49	0.57	0.53	0.53	0.53	0.58	0.73	0.58	0.48	0.60	0.72	0.62
		Weighted	0.90	0.91	0.94	0.92	0.86	0.92	0.92	0.94	0.96	0.97	0.87	0.92	0.92	0.92	0.90	0.93	0.92	0.92
Recall	Class	V_H	0.01	0.33	0.07	0.59	0.01	0.38	0.01	0.31	1.00	1.00	0.29	0.86	0.12	0.44	0.01	0.52	0.05	0.38
		V_L	1.00	0.93	1.00	0.88	1.00	0.97	1.00	0.93	0.55	0.55	0.85	0.65	0.99	0.89	1.00	0.90	1.00	0.94
	Average	Macro	0.50	0.63	0.54	0.74	0.51	0.68	0.51	0.62	0.78	0.78	0.57	0.76	0.56	0.67	0.51	0.71	0.53	0.66
		Weighted	0.94	0.90	0.94	0.86	0.93	0.93	0.56	0.56	0.56	0.56	0.80	0.67	0.94	0.86	0.95	0.88	0.94	0.90
F1-score	Class	V_H	0.02	0.27	0.13	0.37	0.01	0.42	0.01	0.21	0.11	0.11	0.18	0.28	0.19	0.27	0.01	0.31	0.09	0.32
		V_L	0.97	0.94	0.97	0.92	0.96	0.96	0.98	0.95	0.71	0.71	0.89	0.78	0.97	0.92	0.97	0.93	0.97	0.95
	Average	Macro	0.52	0.61	0.69	0.67	0.49	0.69	0.49	0.59	0.63	0.63	0.55	0.65	0.63	0.62	0.49	0.65	0.61	0.64
		Weighted	0.92	0.90	0.94	0.89	0.89	0.92	0.70	0.70	0.71	0.71	0.83	0.78	0.93	0.89	0.92	0.90	0.93	0.91
Confusion	True Positives (TP)		3033	2830	402	484	545	527	321	300	18	18	71	55	252	225	768	689	634	595
Matrix	False Positives (FP)		183	126	27	12	42	26	13	9	0	0	5	1	14	9	42	20	37	24
	False Negatives (FN)		5	208	0	47	0	18	0	21	15	15	13	29	2	29	0	79	2	41
	True Negatives (TN)		6	63	2	17	0	16	0	4	1	1	2	6	2	7	0	22	2	15
False Negative Rate (FNR)			0.00	0.07	0.00	0.09	0.00	0.03	0.00	0.07	0.45	0.45	0.15	0.35	0.01	0.11	0.00	0.10	0.00	0.06
Mean Absolute Error (MAE)			0.06	0.10	0.06	0.11	0.07	0.07	0.04	0.09	0.44	0.44	0.20	0.33	0.06	0.14	0.05	0.12	0.06	0.10

[a] All outputs are defined according to Table 3.2. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] The results are calculated on the sample dataset 010FD, which is a 10% random sample of the full dataset, 100FD, where year distribution and IPC distributions have been stratified. For every output, the number of datapoints corresponds to the total number of complete fields of features, i.e. datapoints with empty number of features because of cleaning or non-existence are not included.

[c] Class V_H represents high value patents and class V_L represents low value patents.

[d] The sample dataset 010FD is split into three sets (maintaining the distribution of the output variable in each): training, validation and the testing set, with a ratio of 98:1:1 respectively.

[e] Θ refers to the classification threshold or decision threshold, where the model converts the probability returned from the model into a class. In the case when $\Theta = 0.50$, the model is optimised for the lowest loss function and highest accuracy. In the case, when $\Theta \neq 0.50$, the threshold is optimised to maximise the macro average F1-score (Lipton et al., 2014; Provost, 2000).

[f] All evaluation metrics of the error-function derivative and their definitions can be found in 4.3.

Table 5.10 Model evaluation on the sample dataset 010FD for generality_t12 by IPC section (technological area)

Output: Generality_t12[a]			Full Sample		IPC Section													
					A		B		C		D		E		F		G		H	
	Class[c]		V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L
	Total[d]		22836	230904	3144	32453	5321	44689	2068	26563	187	2768	793	6881	2039	21003	4771	53768	4513	45112
Datapoints[b]	Training		22382	228594	3082	31806	5215	43798	2026	26034	183	2712	777	6744	1999	20583	4676	52697	4423	44213
	Validation		226	2310	31	322	53	443	21	263	2	28	8	68	20	209	47	533	45	447
	Testing		228	2333	31	325	53	448	21	266	2	28	8	69	20	211	48	538	45	452
Training / Validation																				
Training Loss				0.41		43.49		29.82		52.94		105.30		93.52		61.62		23.66		30.04
Validation Loss				0.41		43.48		29.82		52.94		105.27		93.50		61.62		23.66		30.04
Training Accuracy				0.91		0.90		0.89		0.90		0.56		0.61		0.84		0.92		0.91
Validation Accuracy				0.91		0.91		0.89		0.93		0.67		0.78		0.90		0.92		0.91
Testing																				
Model			a_i	a_ii	b_i	b_ii	c_i	c_ii	d_i	d_ii	e_i	e_ii	f_i	f_ii	g_i	g_ii	h_i	h_ii	i_i	i_ii
Θ[e]			0.50	0.39	0.50	0.44	0.50	0.40	0.50	0.43	0.50	0.48	0.50	0.49	0.50	0.45	0.50	0.41	0.50	0.41
Accuracy[f]			0.91	0.88	0.91	0.87	0.89	0.89	0.93	0.88	0.80	0.82	0.74	0.77	0.91	0.84	0.93	0.92	0.92	0.91
Precision	Class	V_H	0.46	0.35	0.43	0.33	0.33	0.48	0.75	0.30	0.25	0.25	0.25	0.31	0.44	0.28	0.75	0.49	0.67	0.58
		V_L	0.91	0.94	0.92	0.95	0.90	0.92	0.94	0.96	1.00	1.00	0.96	1.00	0.93	0.95	0.93	0.96	0.92	0.92
	Average	Macro	0.70	0.65	0.68	0.64	0.62	0.70	0.85	0.63	0.63	0.63	0.61	0.66	0.69	0.62	0.84	0.73	0.80	0.75
		Weighted	0.87	0.89	0.88	0.89	0.84	0.87	0.92	0.91	0.95	0.95	0.89	0.93	0.89	0.89	0.92	0.92	0.90	0.89
Recall	Class	V_H	0.05	0.42	0.10	0.48	0.02	0.28	0.14	0.48	1.00	1.00	0.75	1.00	0.20	0.50	0.19	0.54	0.13	0.16
		V_L	0.99	0.93	0.99	0.91	1.00	0.96	1.00	0.91	0.79	0.79	0.74	0.74	0.98	0.88	0.99	0.95	0.99	0.99
	Average	Macro	0.52	0.68	0.55	0.70	0.51	0.62	0.57	0.70	0.90	0.90	0.75	0.87	0.59	0.69	0.59	0.75	0.56	0.58
		Weighted	0.91	0.88	0.91	0.87	0.89	0.89	0.57	0.88	0.80	0.80	0.75	0.77	0.91	0.84	0.93	0.92	0.92	0.91
F1-score	Class	V_H	0.09	0.38	0.16	0.39	0.04	0.35	0.24	0.37	0.40	0.40	0.38	0.47	0.28	0.36	0.30	0.51	0.22	0.25
		V_L	0.95	0.93	0.95	0.93	0.95	0.94	0.97	0.93	0.88	0.88	0.84	0.85	0.95	0.91	0.96	0.95	0.95	0.95
	Average	Macro	0.60	0.66	0.60	0.67	0.56	0.66	0.68	0.66	0.74	0.74	0.67	0.75	0.63	0.65	0.69	0.73	0.66	0.65
		Weighted	0.89	0.88	0.89	0.88	0.86	0.88	0.70	0.89	0.87	0.87	0.81	0.84	0.90	0.86	0.92	0.92	0.91	0.90
Confusion	True Positives (TP)		2320	2160	321	295	446	432	265	243	22	21	51	51	206	185	535	511	449	447
Matrix	False Positives (FP)		217	133	28	16	52	38	18	11	0	0	2	0	16	10	39	22	39	38
	False Negatives (FN)		13	173	4	30	2	16	1	23	6	5	18	18	5	26	3	27	3	5
	True Negatives (TN)		11	95	3	15	1	15	3	10	2	2	6	8	4	10	9	22	6	7
False Negative Rate (FNR)			0.01	0.07	0.01	0.09	0.00	0.04	0.00	0.09	0.21	0.19	0.26	0.26	0.02	0.12	0.01	0.05	0.01	0.01
Mean Absolute Error (MAE)			0.09	0.12	0.09	0.13	0.11	0.11	0.07	0.12	0.20	0.18	0.26	0.23	0.09	0.16	0.07	0.08	0.08	0.09

[a] All outputs are defined according to Table 3.2. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] The results are calculated on the sample dataset 010FD, which is a 10% random sample of the full dataset, 100FD, where year distribution and IPC distributions have been stratified. For every output, the number of datapoints corresponds to the total number of complete fields of features, i.e. datapoints with empty number of features because of cleaning or non-existence are not included.

[c] Class V_H represents high value patents and class V_L represents low value patents.

[d] The sample dataset 010FD is split into three sets (maintaining the distribution of the output variable in each): training, validation and the testing set, with a ratio of 98:1:1 respectively.

[e] Θ refers to the classification threshold or decision threshold, where the model converts the probability returned from the model into a class. In the case when $\Theta = 0.50$, the model is optimised for the lowest loss function and highest accuracy. In the case, when $\Theta \neq 0.50$, the threshold is optimised to maximise the macro average F1-score (Lipton et al., 2014; Provost, 2000).

[f] All evaluation metrics of the error-function derivative and their definitions can be found in 4.3.

Table 5.11 Model evaluation on the sample data 010FD for renewal_t4 by IPC section (technological area)

Output: Renewal_t4[a]			Full Sample		IPC Section													
					A		B		C		D		E		F		G		H	
Datapoints[b]	Class[c]		V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L
	Total[d]		361519	57664	46415	9934	57310	11236	32416	5546	2829	636	8844	1996	26633	5129	102110	12979	84962	10208
	Training		354325	56516	45490	9737	56168	11013	31771	5490	2771	624	8667	1956	26102	5027	100078	12720	83270	1005
	Validation		3579	571	460	98	568	111	321	55	29	6	88	20	264	51	1011	129	842	101
	Testing		3615	577	465	98	574	112	324	56	29	6	89	20	267	51	1021	130	850	102
Training / Validation																				
Training Loss				0.06		25.46		18.53		41.34		104.12		86.35		49.16		6.21		9.72
Validation Loss				0.06		25.46		18.53		41.34		104.10		86.34		49.16		6.22		9.72
Training Accuracy				0.86		0.82		0.84		0.85		0.56		0.63		0.82		0.89		0.89
Validation Accuracy				0.86		0.82		0.83		0.85		0.51		0.71		0.83		0.89		0.89
Testing																				
Model			a_i	a_ii	b_i	b_ii	c_i	c_ii	d_i	d_ii	e_i	e_ii	f_i	f_ii	g_i	g_ii	h_i	h_ii	i_i	i_ii
Θ^e			0.50	0.38	0.50	0.41	0.50	0.38	0.50	0.39	0.50	0.49	0.50	0.47	0.50	0.40	0.50	0.35	0.50	0.37
Accuracy[f]			0.86	0.80	0.82	0.78	0.83	0.74	0.85	0.73	0.63	0.63	0.70	0.65	0.83	0.70	0.89	0.77	0.89	0.85
Precision	Class	V_H	0.86	0.30	0.83	0.88	0.84	0.88	0.85	0.88	0.83	0.83	0.83	0.85	0.84	0.85	0.89	0.92	1.00	0.90
		V_L	0.20	0.90	0.50	0.40	0.33	0.30	0.01	0.23	0.18	0.18	0.22	0.25	0.25	0.21	0.01	0.22	0.89	0.23
	Average	Macro	0.53	0.60	0.67	0.64	0.59	0.59	0.43	0.56	0.51	0.51	0.53	0.55	0.55	0.53	0.45	0.57	0.95	0.57
		Weighted	0.77	0.81	0.77	0.80	0.76	0.79	0.73	0.78	0.72	0.72	0.71	0.74	0.75	0.75	0.79	0.84	0.91	0.83
Recall	Class	V_H	1.00	0.37	0.99	0.84	0.99	0.79	1.00	0.80	0.69	0.69	0.80	0.70	0.99	0.77	1.00	0.81	1.00	0.92
		V_L	0.01	0.86	0.05	0.48	0.02	0.47	0.01	0.36	0.33	0.33	0.25	0.45	0.02	0.31	0.01	0.42	0.01	0.19
	Average	Macro	0.51	0.62	0.52	0.66	0.51	0.63	0.51	0.58	0.51	0.51	0.53	0.58	0.51	0.54	0.51	0.62	0.51	0.56
		Weighted	0.86	0.80	0.82	0.78	0.83	0.74	0.85	0.73	0.63	0.63	0.70	0.65	0.83	0.70	0.89	0.77	0.89	0.85
F1-score	Class	V_H	0.93	0.33	0.90	0.86	0.91	0.83	0.92	0.84	0.75	0.75	0.81	0.77	0.91	0.81	0.94	0.86	1.00	0.91
		V_L	0.01	0.88	0.09	0.44	0.04	0.37	0.01	0.28	0.23	0.23	0.23	0.32	0.04	0.25	0.01	0.29	0.02	0.21
	Average	Macro	0.52	0.61	0.58	0.65	0.54	0.61	0.46	0.57	0.51	0.51	0.53	0.56	0.52	0.53	0.48	0.59	0.66	0.56
		Weighted	0.81	0.80	0.79	0.79	0.79	0.76	0.79	0.75	0.67	0.67	0.70	0.69	0.79	0.72	0.84	0.80	0.90	0.84
Confusion	True Positives (TP)		3607	3118	460	392	570	452	324	258	20	20	71	62	264	206	1021	829	850	786
Matrix	False Positives (FP)		575	362	94	51	110	59	56	36	4	4	15	11	50	35	130	76	101	83
	False Negatives (FN)		8	497	5	73	4	122	0	66	9	9	18	27	3	61	0	192	0	64
	True Negatives (TN)		2	215	5	48	2	53	0	20	2	2	5	9	1	16	0	54	1	19
False Negative Rate (FNR)			0.00	0.14	0.01	0.16	0.01	0.21	0.00	0.20	0.31	0.31	0.20	0.30	0.01	0.23	0.00	0.19	0.00	0.08
Mean Absolute Error (MAE)			0.14	0.20	0.18	0.22	0.17	0.26	0.15	0.27	0.37	0.37	0.30	0.35	0.17	0.30	0.11	0.23	0.11	0.15

[a] All outputs are defined according to Table 3.2. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] The results are calculated on the sample dataset 010FD, which is a 10% random sample of the full dataset, 100FD, where year distribution and IPC distributions have been stratified. For every output, the number of datapoints corresponds to the total number of complete fields of features, i.e. datapoints with empty number of features because of cleaning or non-existence are not included.

[c] Class V_H represents high value patents and class V_L represents low value patents.

[d] The sample dataset 010FD is split into three sets (maintaining the distribution of the output variable in each): training, validation and the testing set, with a ratio of 98:1:1 respectively.

[e] Θ refers to the classification threshold or decision threshold, where the model converts the probability returned from the model into a class. In the case when $\Theta = 0.50$, the model is optimised for the lowest loss function and highest accuracy. In the case, when $\Theta \neq 0.50$, the threshold is optimised to maximise the macro average F1-score (Lipton et al., 2014; Provost, 2000).

[f] All evaluation metrics of the error-function derivative and their definitions can be found in 4.3.

Table 5.12 Model evaluation on the sample data 010FD for renewal_t8 by IPC section (technological area)

Output: Renewal_t8[a]			Full Sample		IPC Section													
					A		B		C		D		E		F		G		H	
Datapoints[b]	Class[c]		V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L
	Total[d]		209474	119676	25254	18790	33870	23777	19963	12252	1745	1342	4931	3938	15626	10597	58771	27130	49314	21850
	Training		205305	117294	24750	18416	33196	23303	19565	12008	1727	1316	4833	3859	15314	10386	57601	26589	48332	21415
	Validation		2074	1185	251	186	335	236	198	121	18	13	49	39	155	105	582	269	489	216
	Testing		2095	1197	253	188	339	238	200	123	18	13	49	40	157	106	588	272	493	219
Training / Validation																				
Training Loss				0.17		35.36		24.83		48.91		105.61		91.83		57.58		12.38		17.56
Validation Loss				0.17		35.37		13.83		48.92		105.59		91.81		57.58		12.38		17.56
Training Accuracy				0.67		0.65		0.60		0.63		0.59		0.57		0.59		0.69		0.70
Validation Accuracy				0.66		0.64		0.61		0.61		0.52		0.59		0.61		0.69		0.70
Testing																				
Model			a_i	a_ii	b_i	b_ii	c_i	c_ii	d_i	d_ii	e_i	e_ii	f_i	f_ii	g_i	g_ii	h_i	h_ii	i_i	i_ii
Θ[e]			0.50	0.46	0.50	0.48	0.50	0.46	0.50	0.45	0.50	0.49	0.50	0.49	0.50	0.48	0.50	0.44	0.50	0.45
Accuracy[f]			0.67	0.66	0.65	0.64	0.63	0.64	0.63	0.61	0.55	0.58	0.66	0.63	0.60	0.61	0.69	0.65	0.70	0.68
Precision	Class	V_H	0.68	0.73	0.67	0.69	0.64	0.73	0.65	0.71	0.62	0.67	0.69	0.67	0.64	0.66	0.70	0.75	0.71	0.74
		V_L	0.60	0.53	0.61	0.58	0.60	0.55	0.53	0.49	0.47	0.50	0.63	0.59	0.51	0.52	0.57	0.45	0.53	0.47
	Average	Macro	0.64	0.63	0.64	0.64	0.62	0.64	0.59	0.60	0.55	0.59	0.66	0.63	0.58	0.59	0.64	0.60	0.62	0.61
		Weighted	0.65	0.66	0.64	0.64	0.62	0.65	0.60	0.63	0.56	0.60	0.66	0.63	0.59	0.60	0.66	0.65	0.65	0.66
Recall	Class	V_H	0.89	0.73	0.76	0.68	0.84	0.62	0.89	0.62	0.56	0.56	0.71	0.65	0.79	0.74	0.95	0.74	0.95	0.82
		V_L	0.27	0.52	0.50	0.59	0.33	0.67	0.21	0.59	0.54	0.62	0.60	0.60	0.33	0.42	0.14	0.46	0.13	0.37
	Average	Macro	0.58	0.63	0.63	0.64	0.59	0.65	0.55	0.61	0.55	0.59	0.66	0.63	0.56	0.58	0.55	0.60	0.54	0.60
		Weighted	0.67	0.66	0.65	0.64	0.63	0.64	0.63	0.61	0.55	0.58	0.66	0.63	0.60	0.61	0.69	0.65	0.70	0.68
F1-score	Class	V_H	0.77	0.73	0.71	0.68	0.73	0.67	0.75	0.66	0.59	0.61	0.70	0.66	0.71	0.70	0.81	0.74	0.81	0.78
		V_L	0.38	0.52	0.55	0.58	0.43	0.60	0.30	0.54	0.50	0.55	0.61	0.59	0.40	0.46	0.22	0.45	0.21	0.41
	Average	Macro	0.61	0.63	0.63	0.64	0.60	0.64	0.57	0.60	0.55	0.59	0.66	0.63	0.57	0.58	0.59	0.60	0.58	0.60
		Weighted	0.66	0.66	0.64	0.64	0.62	0.64	0.61	0.62	0.55	0.59	0.66	0.63	0.59	0.60	0.67	0.65	0.67	0.67
Confusion	True Positives (TP)		1874	1538	192	173	286	210	177	124	10	10	35	32	124	116	559	433	467	403
Matrix	False Positives (FP)		868	575	94	78	160	79	97	50	6	5	16	16	71	61	234	146	190	138
	False Negatives (FN)		221	557	61	80	53	129	23	76	8	8	14	17	33	41	29	155	26	90
	True Negatives (TN)		329	622	94	110	78	159	26	73	7	8	24	24	35	45	38	126	29	81
False Negative Rate (FNR)			0.11	0.27	0.24	0.32	0.16	0.38	0.12	0.38	0.44	0.44	0.29	0.35	0.21	0.26	0.05	0.26	0.05	0.18
Mean Absolute Error (MAE)			0.33	0.34	0.35	0.36	0.37	0.36	0.37	0.39	0.45	0.42	0.34	0.37	0.40	0.39	0.31	0.35	0.30	0.32

[a] All outputs are defined according to Table 3.2. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] The results are calculated on the sample dataset 010FD, which is a 10% random sample of the full dataset, 100FD, where year distribution and IPC distributions have been stratified. For every output, the number of datapoints corresponds to the total number of complete fields of features, i.e. datapoints with empty number of features because of cleaning or non-existence are not included.

[c] Class V_H represents high value patents and class V_L represents low value patents.

[d] The sample dataset 010FD is split into three sets (maintaining the distribution of the output variable in each): training, validation and the testing set, with a ratio of 98:1:1 respectively.

[e] Θ refers to the classification threshold or decision threshold, where the model converts the probability returned from the model into a class. In the case when $\Theta = 0.50$, the model is optimised for the lowest loss function and highest accuracy. In the case, when $\Theta \neq 0.50$, the threshold is optimised to maximise the macro average F1-score (Lipton et al., 2014; Provost, 2000).

[f] All evaluation metrics of the error-function derivative and their definitions can be found in 4.3.

Table 5.13 Model evaluation on the sample data 010FD for renewal_t12 by IPC section (technological area)

Output: Renewal_t12[a]			Full Sample		IPC Section													
					A		B		C		D		E		F		G		H	
Datapoints[b]	Class[c]		V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L
	Total[d]		113582	165675	13887	24622	18971	32586	11536	17246	1000	1836	2743	5127	8910	14558	30144	38340	26391	30868
	Training		111321	162378	13610	24131	18593	31937	11306	17079	980	1798	2688	5025	8733	14267	29544	37577	25866	30559
	Validation		1125	1640	138	244	188	323	114	173	10	19	27	51	88	146	298	380	261	309
	Testing		1136	1657	139	247	190	326	116	174	10	19	28	51	89	146	302	383	264	312
Training / Validation																				
Training Loss				0.32		23.87		29.07		53.29		105.66		0.59		62.47		18.85		24.83
Validation Loss				0.33		23.88		29.07		53.29		105.64		0.59		63.46		18.85		24.83
Training Accuracy				0.66		0.66		0.64		0.60		0.59		0.94		0.61		0.64		0.63
Validation Accuracy				0.65		0.67		0.66		0.62		0.65		0.94		0.67		0.65		0.66
Testing																				
Model			a_i	a_ii	b_i	b_ii	c_i	c_ii	d_i	d_ii	e_i	e_ii	f_i	f_ii	g_i	g_ii	h_i	h_ii	i_i	i_ii
Θ[e]			0.50	0.49	0.50	0.49	0.50	0.49	0.50	0.49	0.50	0.29	0.50	0.49	0.50	0.49	0.50	0.49	0.50	0.49
Accuracy[f]			0.67	0.67	0.68	0.67	0.66	0.65	0.60	0.60	0.48	0.66	0.66	0.66	0.69	0.68	0.66	0.66	0.62	0.62
Precision	Class	V_H	0.63	0.69	0.58	0.58	0.58	0.59	0.49	0.49	0.31	0.50	0.52	0.52	0.68	0.67	0.65	0.66	0.60	0.60
		V_L	0.69	0.64	0.71	0.71	0.67	0.66	0.62	0.61	0.62	0.71	0.71	0.71	0.69	0.68	0.66	0.65	0.64	0.63
	Average	Macro	0.66	0.67	0.65	0.65	0.63	0.63	0.56	0.55	0.47	0.61	0.62	0.62	0.69	0.68	0.66	0.66	0.62	0.62
		Weighted	0.67	0.67	0.66	0.66	0.64	0.63	0.57	0.56	0.52	0.64	0.65	0.65	0.69	0.67	0.66	0.66	0.62	0.62
Recall	Class	V_H	0.47	0.83	0.38	0.38	0.24	0.19	0.22	0.17	0.40	0.40	0.43	0.43	0.34	0.29	0.49	0.45	0.54	0.52
		V_L	0.81	0.45	0.84	0.84	0.90	0.92	0.85	0.88	0.53	0.79	0.78	0.78	0.90	0.91	0.79	0.82	0.69	0.71
	Average	Macro	0.64	0.64	0.61	0.61	0.57	0.56	0.54	0.53	0.47	0.60	0.61	0.61	0.62	0.60	0.64	0.64	0.62	0.62
		Weighted	0.67	0.67	0.68	0.68	0.66	0.65	0.60	0.60	0.48	0.66	0.66	0.66	0.69	0.68	0.66	0.66	0.62	0.62
F1-score	Class	V_H	0.54	0.75	0.46	0.46	0.34	0.29	0.30	0.72	0.35	0.44	0.47	0.47	0.45	0.40	0.56	0.54	0.57	0.56
		V_L	0.75	0.53	0.77	0.77	0.77	0.77	0.72	0.25	0.57	0.75	0.74	0.74	0.78	0.78	0.72	0.73	0.66	0.67
	Average	Macro	0.65	0.65	0.63	0.63	0.60	0.59	0.54	0.54	0.47	0.60	0.61	0.61	0.65	0.64	0.65	0.64	0.62	0.62
		Weighted	0.67	0.67	0.67	0.67	0.65	0.64	0.58	0.58	0.50	0.65	0.65	0.65	0.69	0.67	0.66	0.66	0.62	0.62
Confusion	True Positives (TP)		539	510	53	53	45	37	25	20	4	4	12	12	30	26	147	137	143	136
Matrix	False Positives (FP)		315	284	39	40	33	26	26	21	9	4	11	11	14	13	79	70	97	90
	False Negatives (FN)		597	626	86	86	145	153	91	96	6	6	16	16	59	63	155	165	121	128
	True Negatives (TN)		1342	1373	208	207	293	300	148	153	10	15	40	40	132	133	304	313	215	222
False Negative Rate (FNR)			0.53	0.55	0.62	0.62	0.76	0.81	0.78	0.83	0.60	0.60	0.57	0.57	0.66	0.71	0.51	0.55	0.46	0.48
Mean Absolute Error (MAE)			0.33	0.33	0.32	0.33	0.34	0.35	0.40	0.40	0.52	0.34	0.34	0.34	0.31	0.32	0.34	0.34	0.38	0.38

[a] All outputs are defined according to Table 3.2. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] The results are calculated on the sample dataset 010FD, which is a 10% random sample of the full dataset, 100FD, where year distribution and IPC distributions have been stratified. For every output, the number of datapoints corresponds to the total number of complete fields of features, i.e. datapoints with empty number of features because of cleaning or non-existence are not included.

[c] Class V_H represents high value patents and class V_L represents low value patents.

[d] The sample dataset 010FD is split into three sets (maintaining the distribution of the output variable in each): training, validation and the testing set, with a ratio of 98:1:1 respectively.

[e] Θ refers to the classification threshold or decision threshold, where the model converts the probability returned from the model into a class. In the case when $\Theta = 0.50$, the model is optimised for the lowest loss function and highest accuracy. In the case, when $\Theta \neq 0.50$, the threshold is optimised to maximise the macro average F1-score (Lipton et al., 2014; Provost, 2000).

[f] All evaluation metrics of the error-function derivative and their definitions can be found in 4.3.

5.3 Model out-of-sample evaluation by sample size evaluation strategy

We evaluate the forecasting ability of our models using a sample size evaluation strategy (see 4.5 and Fig. 4.13). Firstly, we train, validate and test our models on the full dataset (100FD) (5.3.1), and then on a 10% and 3% random sample of the full dataset, 010FD and 003 respectively (5.3.2).

5.3.1 Full dataset (100FD) results

Table 5.14 presents the results from the deploying the methodology on the full dataset (100FD)[1]. The results for forward citations show a consistency across the different time periods, with an accuracy of 0.99 for T=4 (model a_i), 0.94 for T=8 (model b_i), and 0.88 for T=12 (model c_i). Our results show a significant improvement in accuracy, precision, recall, and F1-score, compared to previous studies summarise in Tables 2.10 and 2.12. This is due to the increase of high patents, V_H, as older patents gain forward citations, which is consistent with the observations by Hall (2005). In addition, the false negative rate appears to be very low, while the number of high value patents V_H increasing with time. This appears to be because of the boundaries becoming clearer between V_H and V_L patents (only high valued patent inventions survive).

The results on the generality_t12, also appear to have a high accuracy 0.91, and an F1-score of 0.61 (model d_i). The model identifies all the relevant instances with a high macro average precision 0.72. Comparing the citations_t12 (models c_i and c_ii) and the generality_t12 (models d_i and d_ii), we observe a similar consistency. These models have a low false negative rate, a high precision and recall for V_L, and a high precision for V_H. We also observe that the F1-score is higher for citations_t12 than generality_t12, which is driven by the higher macro average recall. This is due to noise with the introduction of the diversification of patent classes in the generality index, where sometimes patents are placed in some IPC patent classes with small relevance due to prior art citations.

Our results show that we identify very effectively the V_L patents, with a high precision and recall. This appears to be significant contribution, since the algorithm seems to predict well the low value patents V_L, and identifies with high relevance the high value patents V_H.

[1]Unfortunately, due to the computational resources required to run these sort of algorithms, it was not possible to train, validate and test all output categorical proxies on the full dataset. This is partly the reason we have developed the wide evaluation strategy in 4.5. We focus on the most frequently used output proxies (forward citations, see Table 2.12) and train, validate and test the deep learning algorithm on citations_t4, citations_t8, citations_t12, and generality_index_t12.

Table 5.14 Model evaluation on the full dataset (100FD) per output proxy

Output[a,b]			Citations_t4		Citations_t8		Citations_t12		Generality_t12	
Datapoints[c]	Class		V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L
	Total[d]		39857	4152743	213454	3013091	345582	2215114	229938	2330758
	Training		39063	4070104	209206	2953130	338705	2171033	225363	2284375
	Validation		395	41112	2113	29830	3421	21930	2276	23075
	Testing		399	41527	2135	30131	3456	22151	2299	23308
Training / Validation										
Training Loss				0.00		0.01		0.02		0.02
Validation Loss				0.00		0.01		0.02		0.02
Training Accuracy				0.99		0.94		0.88		0.91
Validation Accuracy				0.99		0.94		0.88		0.91
Testing										
Model			a_i	a_ii	b_i	b_ii	c_i	c_ii	d_i	d_ii
Θ[e]			0.50	0.26	0.50	0.36	0.50	0.48	0.50	0.36
Accuracy[f]			0.99	0.98	0.94	0.91	0.88	0.85	0.91	0.85
Precision	Class	V_H	0.40	0.15	0.61	0.36	0.66	0.46	0.52	0.29
		V_L	0.99	0.99	0.94	0.96	0.89	0.92	0.91	0.94
	Average	Macro	0.70	0.57	0.78	0.66	0.78	0.69	0.72	0.62
		Weighted	0.99	0.99	0.92	0.92	0.86	0.85	0.88	0.88
Recall	Class	V_H	0.03	0.32	0.11	0.44	0.25	0.50	0.06	0.44
		V_L	1.00	0.98	1.00	0.95	0.98	0.91	0.99	0.90
	Average	Macro	0.52	0.65	0.56	0.70	0.62	0.71	0.53	0.67
		Weighted	0.99	0.98	0.94	0.91	0.88	0.85	0.91	0.85
F1-score	Class	V_H	0.06	0.20	0.18	0.40	0.36	0.48	0.11	0.35
		V_L	1.00	0.99	0.97	0.95	0.93	0.91	0.95	0.92
	Average	Macro	0.59	0.61	0.65	0.68	0.69	0.70	0.61	0.64
		Weighted	0.99	0.98	0.92	0.92	0.86	0.86	0.88	0.87
Confusion	True Positives (TP)		41512	40801	29985	28474	21707	20092	23184	20870
Matrix	False Positives (FP)		389	272	1903	1192	2595	1711	2164	1285
	False Negatives (FN)		15	726	146	1657	444	2059	124	2438
	True Negatives (TN)		10	127	232	943	861	1754	135	1014
False Negative Rate (FNR)			0.00	0.02	0.00	0.05	0.02	0.09	0.01	0.10
Mean Absolute Error (MAE)			0.01	0.02	0.06	0.09	0.12	0.12	0.09	0.15

[a] All outputs are defined according to Table 3.2. Class V_H represent high value patents and class V_L represent low value patents. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] Due to limited computational resources, only the following output proxy models have been trained, validated and tested on the full dataset 100FD.

[c] The results are calculated on the full dataset, 100FD, (where year distribution and IPC distributions have been stratified). For every output, the number of datapoints corresponds to the total number of complete fields of features, i.e. datapoints with empty number of features because of cleaning or non-existence are not included.

[d] The full dataset is split into three sets (maintaining the distribution of the output variable in each): training, validation and the testing set, with a ratio of 98:1:1 respectively.

[e] Θ refers to the classification threshold or decision threshold, where the model converts the probability returned from the model into a class. In the case when $\Theta = 0.50$, the model is optimised for the lowest loss function and highest accuracy. In the case, when $\Theta \neq 0.50$, the threshold is optimised to maximise the macro average F1-score (Lipton et al., 2014; Provost, 2000).

[f] All evaluation metrics of the error-function derivative and their definitions can be found in 4.3.

5.3.2 Random representative samples of the full dataset

5.3.2.1 Sample 010FD

Sample dataset 010FD is a 10% random representable sample of the full dataset (100FD)[1]. Table 5.15 shows the results of the models being trained, validated and tested on the 10% random sample (010FD) of the full dataset (100FD), with 98% of 010FD used as training dataset, 1% of 010FD used as validation dataset and 1% of 010FD used as testing dataset[2].

The results show that the model is well trained on the training dataset, validated on the validation dataset and tested on the testing dataset. The loss values for both training and validation are consistent and low for output proxies. We note that the training and validation losses, while low, they are not as low as the 100FD dataset's results (Table 5.14). This is due to the large scale difference in the number of datapoints, which is relative to each model's complexity, resulting in a more consistent adjustment of weight parameters that turned towards very small numbers in order to minimise the loss function. The false negative rate is low for all output proxies, except for renewal_t8 (models i_i and i_ii) and renewal_t12 (models l_i and l_ii), which follows the mean absolute error. Citations_t4 (model d_i and d_ii) has the highest accuracy model, driven by the high number of V_L patents. Quality_index_4 (models c_i and c_ii) has the highest macro average precision and recall, driven by the high precision and recall for V_L. The grant lag (models a_i and a_ii) has an accuracy of 0.77, with a macro average F1-Score of 0.64. This is mainly driven by the low recall of class V_H.

In the short term (t4), citations_t4 model (models d_i and d_ii) has the highest accuracy with 0.99, followed by generality_t4 (models e_i and e_ii) with 0.98 and renewal_t4 (models f_i and f_ii) with 0.86. This is consistent across t4, with macro average F1-scores around 0.50, driven by the low precision and recall values for V_H. All three models can identify the V_L patents, but find it difficult to classify the V_H patents. This partly arises because of the large class imbalance at the early stages of the patent lifecycle, making it difficult to predict the value class in the short term. However, these models are useful because they identify well the V_L patents, and thus can identify a wrong investment into a technology, i.e. a patent that belongs to class V_L, but the model classifies it into class V_H (see 4.3.2). This is also reflected in the low false negative rate.

[1]The 10% random representative sample dataset (010FD) is a 10% random sample of the full dataset (100FD), where year distribution and IPC distribution of patents have been stratified (Brownlee, 2017g, 2020g; Dobbin & Simon, 2011; Ng & Katanforoosh, 2020).

[2]The dataset split and the dataset and variations for training, validation and testing are described and explained in 4.5.2, where the percentage split ensures a representative distribution of the categoric output proxy in the training, validation and testing dataset (Ng & Katanforoosh, 2020).

Table 5.15 Model evaluation on the sample dataset (010FD), per output proxy

Output[a]			Overall[b]						Short Term (t4)						Medium Term (t8)						Long Term (t12)					
			Grant Lag		Generality		Quality Index 4		Citations _t4		Generality_t4		Renewal_t4		Citations _t8		Generality_t8		Renewal_t8		Citations _t12		Generality_t12		Renewal_t12	
Datapoints[c]	Class		V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L
	Total[d]		129598	389431	69174	362916	42658	476376	4015	415426	9612	409829	361519	57664	21202	301497	18906	303793	209474	119676	34089	221984	22836	230904	113582	165675
	Training		127019	381680	67797	355694	41809	466895	3935	407158	9421	401672	354325	56516	20780	295497	18530	297747	205305	117294	33410	217566	22382	228594	111321	162378
	Validation		1283	3856	685	3593	422	4717	40	4113	95	4058	3579	571	210	2985	187	3008	2074	1185	338	2198	226	2310	1125	1640
	Testing		1296	3895	692	3629	427	4764	40	4155	96	4099	3615	577	212	3015	189	3038	2095	1197	341	2220	228	2333	1136	1657
Training / Validation																										
Training Loss				0.04		0.05		0.02		0.04		0.04		0.06		0.15		0.15		0.17		0.42		0.41		0.32
Validation Loss				0.04		0.05		0.02		0.04		0.04		0.06		0.15		0.15		0.17		0.42		0.41		0.33
Training Accuracy				0.78		0.85		0.94		0.99		0.97		0.86		0.94		0.94		0.67		0.88		0.91		0.66
Validation Accuracy				0.77		0.85		0.94		0.99		0.97		0.86		0.94		0.94		0.66		0.88		0.91		0.65
Testing																										
Model			a_i	a_ii	b_i	b_ii	c_i	c_ii	d_i	d_ii	e_i	e_ii	f_i	f_ii	g_i	g_ii	h_i	h_ii	i_i	i_ii	j_i	j_ii	k_i	k_ii	l_i	l_ii
Θ[e]			0.50	0.45	0.50	0.41	0.50	0.38	0.50	0.27	0.50	0.35	0.50	0.38	0.50	0.37	0.50	0.36	0.50	0.46	0.50	0.42	0.50	0.39	0.50	0.49
Accuracy[f]			0.77	0.76	0.85	0.81	0.95	0.93	0.99	0.97	0.98	0.97	0.86	0.80	0.94	0.90	0.94	0.90	0.67	0.66	0.88	0.86	0.91	0.88	0.67	0.67
Precision	Class	V_H	0.58	0.51	0.61	0.44	0.80	0.56	0.01	0.14	0.01	0.34	0.86	0.30	0.59	0.34	0.17	0.23	0.68	0.73	0.64	0.47	0.46	0.35	0.63	0.69
		V_L	0.79	0.84	0.87	0.91	0.95	0.97	0.99	0.99	0.98	0.98	0.20	0.90	0.94	0.96	0.94	0.96	0.60	0.53	0.89	0.91	0.91	0.94	0.69	0.64
	Average	Macro	0.68	0.68	0.74	0.68	0.88	0.77	0.50	0.57	0.50	0.66	0.53	0.60	0.77	0.65	0.55	0.60	0.64	0.63	0.78	0.69	0.70	0.65	0.66	0.67
		Weighted	0.74	0.76	0.83	0.84	0.94	0.94	0.98	0.99	0.95	0.97	0.77	0.81	0.92	0.92	0.90	0.91	0.65	0.66	0.86	0.85	0.87	0.89	0.67	0.67
Recall	Class	V_H	0.26	0.52	0.22	0.57	0.46	0.73	0.01	0.45	0.01	0.17	1.00	0.37	0.10	0.49	0.01	0.33	0.89	0.73	0.24	0.43	0.05	0.42	0.47	0.83
		V_L	0.94	0.84	0.97	0.86	0.99	0.95	1.00	0.97	1.00	0.99	0.01	0.86	1.00	0.93	1.00	0.93	0.27	0.52	0.98	0.93	0.99	0.93	0.81	0.45
	Average	Macro	0.60	0.68	0.60	0.72	0.73	0.84	0.50	0.71	0.50	0.58	0.51	0.62	0.55	0.71	0.50	0.63	0.58	0.63	0.63	0.68	0.52	0.68	0.64	0.64
		Weighted	0.77	0.76	0.85	0.81	0.95	0.93	0.99	0.97	0.98	0.97	0.86	0.80	0.94	0.90	0.94	0.90	0.67	0.66	0.88	0.86	0.91	0.88	0.67	0.67
F1-score	Class	V_H	0.36	0.51	0.32	0.50	0.58	0.63	0.01	0.21	0.35	0.23	0.93	0.33	0.17	0.40	0.02	0.27	0.77	0.73	0.35	0.45	0.09	0.38	0.54	0.75
		V_L	0.86	0.84	0.92	0.88	0.97	0.96	0.99	0.98	0.93	0.98	0.01	0.88	0.97	0.94	0.97	0.94	0.38	0.52	0.93	0.92	0.95	0.93	0.75	0.53
	Average	Macro	0.64	0.68	0.66	0.69	0.79	0.80	0.50	0.63	0.50	0.62	0.52	0.61	0.64	0.68	0.52	0.61	0.61	0.63	0.70	0.68	0.60	0.66	0.65	0.65
		Weighted	0.75	0.76	0.84	0.82	0.94	0.93	0.98	0.98	0.96	0.97	0.81	0.80	0.93	0.91	0.92	0.90	0.66	0.66	0.87	0.85	0.89	0.88	0.67	0.67
Confusion	True Positives (TP)		3643	3262	3532	3126	4714	4524	4153	4047	4099	4068	3607	3118	3643	2812	3033	2830	1874	1538	2174	2057	2320	2160	539	510
Matrix	False Positives (FP)		954	625	540	298	230	116	38	22	95	80	575	362	954	108	183	126	868	575	260	196	217	133	315	284
	False Negatives (FN)		252	633	97	503	50	240	2	108	0	31	8	497	252	203	5	208	221	557	46	163	13	173	597	626
	True Negatives (TN)		342	671	152	394	197	311	2	18	1	16	2	215	342	104	6	63	329	622	81	145	11	95	1342	1373
False Negative Rate (FNR)			0.06	0.16	0.03	0.14	0.01	0.05	0.00	0.03	0.00	0.01	0.00	0.14	0.06	0.07	0.00	0.07	0.11	0.27	0.02	0.07	0.01	0.07	0.53	0.55
Mean Absolute Error (MAE)			0.23	0.24	0.06	0.19	0.05	0.07	0.01	0.03	0.02	0.03	0.14	0.20	0.06	0.10	0.06	0.10	0.33	0.34	0.12	0.14	0.09	0.12	0.33	0.33

[a] All outputs are defined according to Table 3.2. Class V_H represents high value patents and class V_L represents low value patents. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] Outputs in the overall category are constructed and calculated according to Squicciarini et al. (2013).

[c] The results are calculated on the sample dataset 010FD, which is a 10% random sample of the full dataset, 100FD, where year distribution and IPC distributions have been stratified. For every output, the number of datapoints corresponds to the total number of complete fields of features, i.e. datapoints with empty number of features because of cleaning or non-existence are not included.

[d] The sample dataset 010FD is split into three sets (maintaining the distribution of the output variable in each): training, validation and the testing set, with a ratio of 98:1:1 respectively.

[e] Θ refers to the classification threshold or decision threshold, where the model converts the probability returned from the model into a class. In the case when $\Theta = 0.50$, the model is optimised for the lowest loss function and highest accuracy. In the case, when $\Theta \neq 0.50$, the threshold is optimised to maximise the macro average F1-score (Lipton et al., 2014; Provost, 2000)

[f] All evaluation metrics of the error-function derivative and their definitions can be found in 4.3.

Table 5.16 Model evaluation on the sample dataset (003FD), per output proxy

Output[a]			Overall[b]						Short Term (t4)						Medium Term (t8)						Long Term (t12)					
			Grant Lag		Generality		Quality Index 4		Citations _t4		Generality_t4		Renewal_t4		Citations _t8		Generality_t8		Renewal_t8		Citations _t12		Generality_t12		Renewal_t12	
Datapoints[c]	Class		V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L	V_H	V_L
	Total[d]		38754	116955	20659	109108	12912	142798	1180	124545	2934	122791	108342	17334	6336	90240	5737	90839	62605	35879	10148	66649	6928	69869	34100	49560
	Training		37982	114627	20247	106937	12655	139955	1156	122066	2876	120346	106185	16989	6210	88443	5623	89030	61359	35165	9946	65322	6790	68478	33421	48573
	Validation		384	1158	205	1080	128	1414	12	1233	29	1216	1073	172	63	894	57	900	620	355	101	660	69	692	338	491
	Testing		384	1170	207	1091	129	1429	12	1246	29	1229	1084	173	63	903	57	909	626	359	101	667	69	699	341	496
Training / Validation																										
Training Loss				2.72		4.54		2.61		4.84		4.86		4.97		9.36		9.39		9.28		20.05		26.58		13.03
Validation Loss				2.72		4.54		2.61		4.84		4.85		4.97		9.36		9.39		9.28		20.05		26.58		13.03
Training Accuracy				0.76		0.85		0.94		0.99		0.98		0.86		0.94		0.94		0.66		0.88		0.91		0.64
Validation Accuracy				0.77		0.85		0.94		0.99		0.98		0.86		0.94		0.94		0.66		0.89		0.91		0.62
Testing																										
Model			a_i	a_ii	b_i	b_ii	c_i	c_ii	d_i	d_ii	e_i	e_ii	f_i	f_ii	g_i	g_ii	h_i	h_ii	i_i	i_ii	j_i	j_ii	k_i	k_ii	l_i	l_ii
Θ^e			0.50	0.44	0.50	0.40	0.50	0.46	0.50	0.32	0.50	0.34	0.50	0.37	0.50	0.38	0.50	0.39	0.50	0.47	0.50	0.40	0.50	0.38	0.50	0.48
Accuracy[f]			0.76	0.74	0.85	0.85	0.95	0.95	0.99	0.98	0.98	0.96	0.86	0.76	0.94	0.90	0.94	0.93	0.65	0.65	0.88	0.85	0.91	0.88	0.66	0.65
Precision	Class	V_H	0.54	0.47	0.56	0.43	0.82	0.76	0.01	0.20	0.01	0.23	0.86	0.88	0.57	0.34	1.00	0.35	0.67	0.70	0.62	0.44	0.67	0.35	0.63	0.62
		V_L	0.77	0.82	0.87	0.92	0.95	0.96	0.99	0.99	0.98	0.98	0.01	0.23	0.94	0.97	0.94	0.95	0.55	0.53	0.89	0.93	0.91	0.94	0.68	0.66
	Average	Macro	0.66	0.65	0.72	0.68	0.89	0.86	0.50	0.60	0.50	0.61	0.44	0.56	0.76	0.66	0.97	0.65	0.61	0.62	0.76	0.69	0.79	0.65	0.66	0.64
		Weighted	0.71	0.73	0.82	0.84	0.94	0.94	0.98	0.99	0.95	0.97	0.74	0.79	0.91	0.93	0.95	0.92	0.62	0.64	0.86	0.87	0.89	0.89	0.66	0.65
Recall	Class	V_H	0.13	0.46	0.20	0.62	0.46	0.50	0.01	0.25	0.01	0.31	1.00	0.84	0.06	0.56	0.02	0.23	0.90	0.80	0.23	0.57	0.06	0.41	0.43	0.38
		V_L	0.96	0.83	0.97	0.85	0.99	0.99	1.00	0.99	1.00	0.97	0.01	0.30	1.00	0.93	1.00	0.97	0.21	0.39	0.98	0.89	1.00	0.92	0.82	0.84
	Average	Macro	0.55	0.65	0.59	0.74	0.73	0.75	0.51	0.62	0.51	0.64	0.51	0.57	0.53	0.75	0.51	0.60	0.56	0.60	0.61	0.73	0.53	0.67	0.63	0.61
		Weighted	0.76	0.74	0.85	0.81	0.95	0.95	0.99	0.98	0.98	0.96	0.86	0.76	0.94	0.90	0.94	0.93	0.65	0.65	0.88	0.85	0.91	0.88	0.66	0.65
F1-score	Class	V_H	0.21	0.46	0.30	0.51	0.59	0.60	0.01	0.22	0.01	0.26	0.93	0.86	0.11	0.42	0.03	0.28	0.77	0.74	0.33	0.50	0.11	0.37	0.51	0.47
		V_L	0.86	0.83	0.91	0.88	0.97	0.97	0.99	0.99	0.99	0.98	0.01	0.26	0.97	0.95	0.97	0.96	0.33	0.45	0.93	0.91	0.95	0.93	0.74	0.74
	Average	Macro	0.59	0.65	0.64	0.70	0.80	0.80	0.50	0.61	0.50	0.62	0.47	0.56	0.62	0.70	0.67	0.62	0.58	0.60	0.67	0.71	0.63	0.65	0.64	0.62
		Weighted	0.69	0.74	0.82	0.82	0.94	0.94	0.99	0.98	0.97	0.96	0.80	0.78	0.94	0.91	0.91	0.92	0.60	0.64	0.86	0.86	0.88	0.88	0.65	0.63
Confusion	True Positives (TP)		1126	971	1058	1009	1416	1408	1245	1234	1229	1198	1083	908	900	836	909	885	563	501	653	593	697	646	146	130
Matrix	False Positives (FP)		337	211	165	118	70	64	12	9	29	20	173	121	59	28	56	44	282	218	78	43	65	41	87	78
	False Negatives (FN)		44	199	33	82	13	21	0	12	0	31	1	176	3	67	0	24	63	125	14	74	2	53	195	211
	True Negatives (TN)		51	177	42	89	59	65	1	3	0	9	0	52	4	35	1	13	77	141	23	58	4	28	409	418
False Negative Rate			0.04	0.17	0.03	0.08	0.01	0.01	0.00	0.01	0.00	0.03	0.00	0.16	0.00	0.07	0.00	0.03	0.10	0.20	0.02	0.11	0.00	0.08	0.57	0.62
Mean Absolute Error			0.24	0.26	0.15	0.15	0.05	0.05	0.01	0.02	0.02	0.04	0.14	0.24	0.06	0.10	0.06	0.07	0.35	0.35	0.12	0.15	0.09	0.12	0.34	0.35

[a] All outputs are defined according to Table 3.2. Class V_H represents high value patents and class V_L represents low value patents. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[b] Outputs in the overall category are constructed and calculated according to Squicciarini et al. (2013).

[c] The results are calculated on the sample dataset 003FD, which is a 3% random sample of the full dataset, 100FD, where year distribution and IPC distributions have been stratified. For every output, the number of datapoints corresponds to the total number of complete fields of features, i.e. datapoints with empty number of features because of cleaning or non-existence are not included.

[d] The sample dataset 003FD is split into three sets (maintaining the distribution of the output variable in each): training, validation and the testing set, with a ratio of 98:1:1 respectively.

[e] Θ refers to the classification threshold or decision threshold, where the model converts the probability returned from the model into a class. In the case when $\Theta = 0.50$, the model is optimised for the lowest loss function and highest accuracy. In the case, when $\Theta \neq 0.50$, the threshold is optimised to maximise the macro average F1-score (Lipton et al., 2014; Provost, 2000).

[f] All evaluation metrics of the error-function derivative and their definitions can be found in 4.3.

In the medium term (t8), both citations_t8 (models g_i and g_ii) and generality_t8 (models h_i and h_ii) have an accuracy of 0.94, where as renewal_t8 (models i_i and i_ii) drops to 0.67. The number of V_H patents increases, and thus the model are able to identify them. This is reflected in the higher precision both in V_H and the macro average, while maintaining the false negative rate low. Recall for V_L is very high indicating that these models are able to maintain the identification of the majority class from t4. Renewal_t8 has an accuracy of 0.67, and an F1-score of 0.61. However, the false negative rate increases, partly driven by the rise in false negatives. As we increase the forecasting window, the model finds it more difficult to identify which patents are renewed and thus are valuable. This reinforces the question whether firms do renew their most valuable patents, or they renew the majority of their patent portfolios in fear of losing out.

In the long term (t12), both citations_t12 (models j_i and j_ii) and generality_t12 (models k_i and k_ii) are close with accuracies of 0.88, and 0.91 respectively, while renewal_t12 (models l_i and l_ii) remains at 0.67. While the macro average F1-scores rise due to the number of V_H patents increasing, the false negative rate remains low, indicating that the models are able to better identify V_H and V_L patents, with higher macro average precision. These models are less able to distinguish between true negatives and false positives, and that is why the number of false positives increases. In addition, for renewal_t12 (models l_i and l_ii), the false negative rate increases to 0.53, which indicates a high number of false positives and false negatives, strengthening the above that firms decide to renew all of their patents and not in a strategic manner.

5.3.2.2 Sample 003FD

Sample dataset 003FD is a 3% random representable sample of the full dataset (100FD)[1]. Table 5.16 shows the results of the models being trained, validated and tested on the 3% random sample of the full dataset (100FD), with 98% of 003FD used as training dataset, 1% of 003FD used as validation dataset and 1% of 003FD used as testing dataset[2].

The results show that the model is well trained on the training set, validated on the validation set and tested on the testing set. The loss for both training and validation is not as low as the results in the 010FD sample dataset (Table 5.15) or the 100FD full dataset (Table 5.14). This implies that while the model's error loss function (see 4.4.2.2) reaches

[1]The 3% random representative sample dataset (010FD) is a 3% random sample of the full dataset (100FD), where year distribution and IPC distribution of patents have been stratified (Brownlee, 2017g, 2020g; Dobbin & Simon, 2011; Ng & Katanforoosh, 2020).

[2]The dataset split and the dataset and variations for training, validation and testing are described and explained in 4.5.2, where the percentage split ensures a representative distribution of the categoric output proxy in the training, validation and testing dataset (Ng & Katanforoosh, 2020).

the minimum, the weight parameters are not fully adjusted and thus keeping the error at a higher than anticipated value. This is also supported by the increase in the false negative rate relative to Table 5.15. Citations_t4 model (models d_i and d_ii) has the highest accuracy, driven by the high number of V_L patents[1].

In the short term (t4), citations_t4 model (models d_i and d_ii) has an accuracy of 0.99, followed by generality_t4 (models e_i and e_ii) with 0.98 and renewal_t4 (models f_i and f_ii) with 0.86. In addition, we observe that the classification thresholds for the adjusted models (models d_ii, e_ii and f_ii) lie between 0.32-0.37, which shows a consistency in the identification of the boundary between V_H and V_L, but a difficulty in correctly classifying the V_H patents. With the decision boundary optimisation, we can clearly observe an increase in recall for all three output proxies, mainly driven by the increase in recall of V_H patents (Provost, 2000).

In the medium term (t8), both the citations_t8 model (models g_i and g_ii) and the generality_t8 model (models h_i and h_ii) have an accuracy of 0.94, where as the renewal_t8 model (models i_i and i_ii) drops to 0.57. In the long term (t12), both the citations_t12 model (models j_i and j_ii) and the generality_t12 model (models k_i and k_ii) are close with accuracies of 0.88, and 0.91 respectively, while the renewal_t12 model (models l_i and l_ii) remains at 0.66. These results follow the results of Table 5.15 dataset, with higher training and validation errors and false negative rates, despite the changing parameter which is the size of the dataset (because the model complexity is fixed). If a firm is interested in monitoring technologies, then models trained on the 3% sample (003FD), are more suitable since speed is of essence. Firms interested in investing in technologies and technology development should focus more on the models with a higher number of datapoints such as the 10% sample (010FD) and the full dataset (100FD), where the error is lower, false negative rates are lower, and accuracy and F1-score are higher.

[1]When we optimise the classification decision threshold to increase the macro average F1-score, the citation_t4 model reaches 0.61 with a threshold of 0.32, which is quite close to the 0.63 from the 010FD sample dataset and threshold of 0.27 (Table 5.15).

Chapter 6

Discussion

The analysis of the literature (chapter 2), reveals that there are limited studies of applying artificial intelligence (AI) methodologies for valuing patents, predominantly using small sample sizes (see 2.1.2.6 and Tables 2.11 and 2.12). Only a limited number of studies make use of artificial neural networks (ANN), with the majority of the studies using low capacity[1] shallow ANNs[2], with binary classification, limited numeric and categoric features, and one output variable. Thus, there is a limited number of studies using deep learning[3] for the valuation of patents.

In this chapter, we provide an interpretation of the results transparently reported in chapter 5, from the deployment of our developed deep learning algorithmic approach (chapter 4), referring to prior literature using examples. We identify patterns from some of the observed trend explanations provided in chapter 5, and refer them back to the literature. Given the limitations of previous research, we discuss two main topics: (i) valuation of patents using AI methodologies (6.1), and (ii) the advancement of AI methodologies for the analysis of patent data and deployment for patent valuation purposes (6.2).

Firstly, we focus on discussing the valuation of patented inventions referring to prior

[1]The capacity of a neural network is defined as configuration of neurons or nodes and layers, i.e. the number of layers, the number of input nodes, the number of output nodes, and the number of nodes in each layer (Hopfield, 1982; Jia et al., 2016), and controls the scope of the types of mapping functions that it is able to learn (Brownlee, 2019g).

[2]Deep neural networks are defined as networks with architectures with multiple hidden layers, where as shallow neural networks have one or two hidden layer (Delalleau & Bengio, 2011; Goodfellow et al., 2016; Murphy, 2012).

[3]For the purpose of this research (see 2.2), we use the term *deep learning (DL)* to describe artificial neural networks (ANN), in supervised learning paradigms, defined by the depth of the credit assignment paths, which are chains of possibly learnable, causal links between inputs and outputs (Hinton et al., 2006), i.e. finding weights that make the neural network exhibit desired behaviour (Schmidhuber, 2015). These are also known as deep (and wide) neural networks (Cheng et al., 2017; Goodfellow et al., 2016; Shaked et al., 2016). Brownlee (2019m), URL: https://machinelearningmastery.com/what-is-deep-learning/.

literature in 6.1, relative to the following: (i) patent value output proxies deployed (6.1.1); (ii) composite indices deployed for patent value (6.1.2); and (iii) the value dimension the model outputs represent and how these could partly be interpreted (6.1.3). Secondly, we focus on the AI methodologies deployed for patent valuation in 6.2, discussing the following: (i) the advancements of the deep learning approach in the analysis of patent data for forecasting patent value (6.2.1); (ii) the deployment of our deep learning approach to forecast patent value for a variety of technological areas (6.2.2); and (iii) the advancements of the deep learning approach to forecast patent value by using the patent text (abstract, claims, title, summary) (6.2.3).

6.1 Valuation of patented inventions

6.1.1 Patent value output proxies

As the literature reveals (chapter 2), this research is one of few to explore a range of patent value output proxies using artificial intelligence (AI) methodologies and large datasets. This is in contrast to current research, i.e. Lee et al. (2018) and Noh & Lee (2020), who explicitly use only a few lagging proxies, such as forward citations (see 2.2.2.3.2.1 and Table 2.12). Most of prior studies depend on forward citations, making the analysis of newly filed patents with little or no forward citations practically impossible (Jun & Lee, 2012). Thus, relying only on forward citations as a patent value proxy could lead to incomplete patent value assessment. Woo et al. (2019) use the generality output proxy with a very low macro average F1-scores. In contrast, from Table 5.15, we find a macro average optimised F1-scores (i.e. $\Theta \neq 0.50$) of 0.62 for generality_4 and generality_8, and 0.66 for generality_12. We observe that the generality proxy, which is the technological diversification of forward citations, improves with increasing time horizon (Hagedoorn & Cloodt, 2003; Leten et al., 2007).

Our results show a significant improvement from prior art. With higher accuracies and macro average F1-scores (see Table 5.14 and Table 5.15), our models are well trained with low values for the training and validation losses (Zhang et al., 2012). They are able to distinguish V_H and V_L better than previous research (Table 2.12). This is driven by the 2-class approach of structuring the problem (see 4.1), relative to the 4-class approach followed by Lee et al. (2018) (see 2.2.2.3.2.4)[1].

For our models, a true positive (TP) is a V_L patent that is predicted to be V_L, and a true negative (TN) is a V_H patent that is predicted to be V_H (see 4.3.2). A false positive (FP) is

[1] Studies in Table 2.12 are able to classify low value patents really well, but are hardly able to find any of the high value patents.

a V_H patent that is predicted to be V_L, i.e. it is a missed opportunity and the patent remains unexploited because the firm's management decides against exploiting it due to the inaccurate prediction (Baglieri & Cesaroni, 2013; Gregory, 1995). A false negative is a V_L patent that is predicted to be V_H, i.e. it is a wrong investment (Arora et al., 2008; Ernst, 1995; Soenksen & Yazdi, 2016; Verbano & Nosella, 2010). The patent is heavily exploited with resource commitment and development investment due to the inaccurate prediction, with more serious implications for firms, leading to financial losses[1].

Moreover, as the prediction horizon increases from t4 to t12, we observe an increase in the macro average F1-scores, because the number of V_H patents tends to increase. This is particularly important comparing our results to Lee et al. (2018) and Woo et al. (2019) (see Table 2.12), since we observe that our models are able to more clearly distinguish the features of V_H relative to V_L patents, i.e. the number of false positives (FP) decreases, while the number of true negatives (TN) increases (Hall et al., 2005).

Trappey et al. (2019) propose a patent value deep learning-based analysis with numerical features, specific for the Internet of Things (IoT). They use a 4-class output proxy to determine if a patent would become a standard essential patents (SEPs) under the Paris Convention Treaty (PCT). They find a training accuracy of 0.83 and testing accuracy of of 0.73. This suggests that there is substantial model overfitting. They also observe an improvement in the testing accuracy when trying a deep neural network architecture, consistent with our methodology development (see 4.4.1.1) on the basis of deep and wide neural networks (Lee & Hsiang, 2019a; Noh & Lee, 2020). However, Choi et al. (2020) achieve a higher macro average F1-score using the constructed proxy of core business patent (CBP). This is based on the patent lifetime renewal, which reduces the scope of the research and focus solely on the strategic and economic value dimensions (see Table 2.3). Their results are driven by the low number of input features together with using a balanced dataset training approach, with equal class ratio. This is in contrast to AI, machine and deep learning research, which suggests to use a cost-sensitive loss function, i.e. focal loss (see 4.10), before using other approaches[2] for a balanced dataset, such as oversampling or undersampling (Altini, 2015; Mao, 2019; Seif, 2018). Harhoff et al. (1999) find that patents renewed full term are more likely to be cited, identifying the noisy relationship between citations and the value distribution. We also observe this in our results with the macro average F1-score increasing for citations_t12, despite a slight drop in the accuracy, because the number of V_H patents tends to increase.

[1]The model is penalised more by the focal loss (see 4.10) when it makes a mistake for a false negative relative to other types of misclassification, with low number of false negatives and low false negative rate (chapter 5) because of the more serious implications that could result for firms with wrong investments.

[2]Wu & Radewagen (2017), URL: https://www.kdnuggets.com/2017/06/7-techniques-handle-imbalanced-data.html.

Our study is one of the very few to explore a wide range of patent value proxies, with large datasets, with an improvement in accuracy, macro average F1-scores and false negative rate relative to prior art. We use 12 output proxies, in three time horizons, with several of our trained models able to distinguish the features of V_H relative to V_L patents (Hall, 2005). This has implications for patent value forecasting using deep learning AI methodologies. Identifying valuable patents using more than one output proxy, based on a number of patent value dimensions, could improve the completeness of the patent value assessment, and subsequently of technologies.

6.1.2 Composite indices of patent value proxies

This research is one of the few in the field (see Table 2.12) to explore the use of composite indices as a patent value proxy using AI methodologies and large datasets. We use the quality_index_4, defined by Squicciarini et al. (2013), which builds on Lanjouw & Schankerman (2004). The index is based on four components: the number of forward citations (up to 5 years after publication), family size, number of claims, and the generality index. Patent quality is a controversial subject, mainly because of the many definitions for the term 'quality' (Aristodemou & Tietze, 2017b). This is a complex and multi-dimensional issue, which suffers from the typical drawback of all composite indicators, and should be interpreted with care (OECD, 2008; Squicciarini et al., 2013). We argue that our index is a patent value proxy, capturing the value of patents, from the technological, economic and strategic value dimension. This is based on Table 2.3, and the value dimensions covered by the components of the index (Poege et al., 2019). Forward citations, the generality_index and the number of claims are associated with the technological and economic value of patents, whereas the patent family size is associated with the economic and strategic value (see Tables 2.4 and Table 2.5).

Table 5.15 shows that the quality_index_4 model has one of the highest accuracies of 0.95 (model c_i, $\Theta = 0.50$) and 0.93 (model c_ii, $\Theta \neq 0.50$) respectively, together with the highest macro average F1-scores of 0.79 (model c_i, $\Theta = 0.50$) and 0.80 (model c_ii, $\Theta \neq 0.50$) respectively. These are driven by the high precision and recall values for the V_H patents, which are consistently high for the V_L patents. We argue that this occurs because of a number of reasons. Firstly, with 0.45, the cut-off threshold between class V_H and V_L is quite high, which positions the high value patents above at least the 75th percentile of patents per year and above at least the 90th percentile of the overall distribution according to Squicciarini et al. (2013). Secondly, drawing on multiple components, there is an inclusive agreement between these proxies, of which are the features of V_H patents relative to V_L patents. Three out of the four components draw mainly on the technological and economic

value dimensions, while the patent family size draws also on the strategic value dimension. This is supported by the argument by Harhoff et al. (2003) that relying on one output proxy alone, specifically forward citations, is not likely to lead to the best possible approximation of the value of patents (Lagrost et al., 2010). Lanjouw & Schankerman (2004) also shows that there is a substantial information gain from using the composite index because of the greater percentage decrease in the variance. Strengthening this, the boundary threshold of the quality_index_4, i.e. the features that distinguish between V_H and V_L patents, increases, and the model is able to understand the distinction between them much better, because these features draw on multiple combinations of the four components (Grimaldi et al., 2015, 2018).

Our study is one of the very few to explore the use of composite indices as patent value proxies with AI methodologies, with large datasets. We find that using a composite index, which draws on multiple components, our models identify the features between V_H and V_L patents better, with improved accuracy and macro average F1-scores (Harhoff et al., 2003; Lagrost et al., 2010). Further research could focus on exploring the use of composite indices for patent valuation with AI, and construction of indices, taking into consideration the patent value dimensions, and other data sources.

6.1.3 Value dimension of patents

This research explores a wide range of patent value output proxies and their associated patent value dimension (see Table 2.3). Frietsch et al. (2010) is one of the few studies that offers a taxonomy for the patent value concepts and identify different motives and incentives for applicants and inventors to file patents (Ernst, 2001). We observe a number of relevant associations of the patent value proxies and their dimensions (see Fig. 6.1).

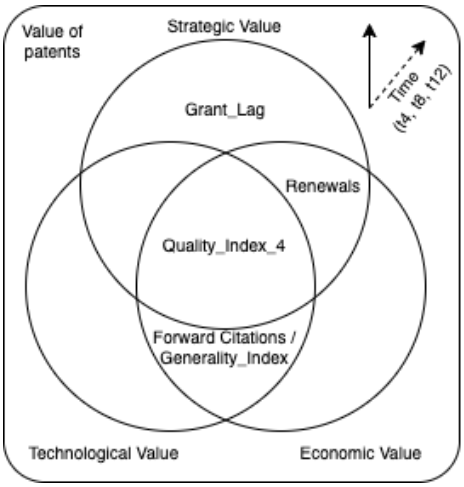


Fig. 6.1 Value dimension of patents and patent value output proxies (based on Table 2.3)

The economic value is defined as the degree to which a patent enters or creates a new market (Frietsch et al., 2010; Lagrost et al., 2010). Forward citations, generality and renewals are all associated with it. Forward citations and the generality_index (see Table 5.15 and Table 5.16) have similar accuracies and macro average F1-scores. This can be partly explained by the overlap in the definition of the generality_index. However, with increasing time horizon, the citations models perform slightly better than the generality_index models, which look at the technology diversification, i.e. the distribution of the forward citations in different IPC classes. This is partly driven by the higher increase in V_H patents for citations than the generality_index, meaning that while forward citations increase, they are more concentrated on related fields. This partially contradicts Chen & Chang (2010d), who argue that firms should diversify their patents or technological capabilities if they want to enhance their market value. Firms with wider technological competencies, have higher probabilities to take advantage of new technological opportunities, and thus have a lower risk of missing new opportunities in the short term.

Moreover, patents that are renewed full term are more likely to be cited and be part of a valuable technology (Harhoff et al., 1999). Deng (2007) empirically 'prove' that several patent owners of high valued inventions not only choose to keep their patents alive longer in a country, but also seek patent protection in more countries (Liu et al., 2008). We also observe this from our results of renewals, with increasing time horizon, the performance of the renewals models drops (see Table 5.1).

The technological value is defined as the degree to which a patent contributes to further developing advanced technology (Chandra & Dong, 2018; Frietsch et al., 2010). Forward citations, and the generality_index, are a measure of technological importance and thus technological value (Albert et al., 1991; Aristodemou & Tietze, 2018a). From our analysis, we observe a similarity between these two output proxies. As the time horizon increases, the models of citation_t12 (see Table 5.15) identify more easily V_H patents. Lanjouw & Schankerman (2001) argue that a greater number of citations implies a greater level of competition on that technology, which we observe above. However, the number of V_H patents in citations_12 grows faster than the number in the generality_index_12, indicating that the forward citations are more concentrated in the same technology classes. This is different for the short term (t4) time horizon where the opposite seems to be happening. This implies that firms file core patents in a diversified number of IPC classes in the short term, and then once these patents are established firms concentrate their resources and new fillings around these core patents in similar IPC classes (Hall et al., 2005; Hall & MacGarvie, 2010).

Patents can create strategic benefits, where patent strategies of innovative companies

become broader and more complex, thus resulting in an expansion of patents[1]. The strategic value is defined as the degree to which a patent is used with an underlying strategic motives to, but not limited to, blocking competitors, easier access to financial markets, preventing key technologies from being invented around and the generation of licensing revenues (Frietsch et al., 2010; Granstrand, 1999). The analysis of patent renewals could serve as a rough estimate of the strategic value of patents (Blind et al., 2006; Harabi, 1995). For example, blocking patents seems to have no direct technological value, yet it has a strategic value (Reitzig, 2004).

As innovation cycles become shorter, the 4 year and 8 year renewal periods suffice to deter market entrants and competitors from patenting in the same field. So patents that are withdrawn in the short term could roughly be seen as an indicator for strategic patenting (Frietsch et al., 2010). However, also given that the cost of maintaining a patent portfolio is not massive, firms seem to continue paying the renewal fees (Blind et al., 2009; Striukova, 2007). This partly explains the performance decrease for the renewal_t12 models (see Table 5.15). With improvements in the digital landscape, reduced search times and administrative burden and the changing competitive landscape has forced many firms to be precise with their patents and look for a quick and robust granting process of the V_H patents (Squicciarini et al., 2013). This is partly observed in the results for grant_lag (see Table 5.1).

All the three value dimensions, economic, technological and strategic, can be represented together by the quality_index_4[2] (see 6.1.2). Table 5.15 shows that the results of the quality_index_4 models are higher than the rest of the output proxies for both accuracy and macro average F1-score[3]. Drawing on multiple value dimensions, there is an inclusive agreement between the elemental proxies that form the composite index, as to which are V_H patents relative to V_L patents. This is also supported by Harhoff et al. (2003) who argue that one output proxy alone is not likely to lead to a representative approximation of patent value (European Commission, 2004; Lagrost et al., 2010; Lanjouw & Schankerman, 2004).

Our study is one of the very few to explore a wide range of patent value proxies and their associated patent value dimensions with AI methodologies, with large datasets. We use 12 patent value proxies, which are represented by 3 dimensions in the literature, economic, strategic and technological. This has several research and practical implications, with firms

[1]The patent system, whose original purpose was to provide a temporally limited protection for technological knowledge seems to be more intensively used by applicants for various strategic motives (Blind et al., 2009; Miller, 2006).

[2]The quality_index_4 is a composite value index, and is based on forward citations, patent family size, number of claims, and the generality index. This inherently means that it represents a complex collective and multi-dimensional value (Squicciarini et al., 2013).

[3]Given the complexity of our models, the complex nature of the output proxies, and the complex nature of the value dimensions these proxies represent, the results should be interpreted with care.

choosing to focus on specific value dimensions, and use certain proxies associated with that dimension, thus reducing time and resources. Moreover, researchers can focus on the less well developed strategic value dimension, exploring the motives that drive renewals and technological diversification in different time horizons, with AI methodologies.

6.2 Methodologies deployed for the value of patents

6.2.1 Patent value methodologies

Table 2.7 reveals that are limited studies with a large number of datapoints that use categoric and numeric data. Table 2.12 shows that there 8 articles that use artificial neural network methodologies with patent data for patent value. We differentiate ourselves from the remaining methodologies, by drawing on several artificial intelligence methodologies. More specifically, this study is one of the very few studies to have used deep and wide artificial neural networks (ANN) of multi-layer perceptrons (MLP), i.e. deep learning[1]. Doing so, we observe improvements in the overall results and the distinction classification between V_H and V_L patents due to the increase in the depth and width of the neural network. Our models use numeric, categoric and text data, with a 7-layer deep neural network, which balances the gains in overall performance, and is able to extract enough information from the data to make a reasonable classification.

We are able to run our models on extensively large datasets, which are not depended on a technology field (see 5.2). With this wide applicability[2] our models reach saturation, with the addition of the batch normalisation and L2 regularisation parameters with low loss function errors. This seems to be mainly because of the comparative number of the dataset's datapoints and the model complexity, allowing our models to identify many relationships between the determinants and proxies. Future research directions can focus on the exploration of variations of deeper ANNs to identify relationships between determinants and proxies.

6.2.2 Technologies

Our approach has allowed us to move away from relatively small datasets, limited to a specific technological field, which have less reproducibility in other fields (see 2.2.2.3.2 and Table 2.12). Models confined to a particular area of technology reflect the characteristics

[1]Earlier studies have used 2 or 3-layer neural networks, with a limited number of numeric or categoric data (see 2.2.2.3.2).

[2]All the patent value determinants are defined ex-ante, at the point or just after the patent has been filed (Lee et al., 2018; Noh & Lee, 2020; van Zeebroeck & van Pottelsberghe de la Potterie, 2011b).

of that area with a high performance, but it may be difficult to apply these models to other fields, and the experiments must be repeated for each field of application[1]. Table 6.1 shows the comparative evaluation of model performance for technology IPC classes. We observe consistently that models trained on IPC G and H, are in the top 3 rankings for all except the generality index output proxy. For forward citations and renewals, this consistency can be due to the high ratio of V_H to V_L patents, which allows the models to identify the features of V_H of patents, and proximity of the language.

Table 6.1 Evaluation of Model Performance for the Technology IPC area

Technology[a]	Overall[b]				Citations			Generality			Renewal		
IPC[c, d]	Grant Lag	Generality	Quality Index 4		Citations_t4	Citations_t8	Citations_t12	Generality_t4	Generality_t8	Generality_t12	Renewal_t4	Renewal_t8	Renewal_t12
A	2	+	1		2	1	1	x	2		1	3	
B					1 , +	2 , +			1 , x , +	x	2		
C		x	x					1					
D								3		1			3
E	1 , x	2								2			
F							+	+					1
G	+	1	2 , +		x	x	2 , x	2	3	3	3	2	2
H	3	3	3		3	3	3			+	x , +	1 , x , +	x , +

[a] Technologies are grouped according to the International Patent Classification (IPC) (Squicciarini et al., 2013), which has been defined by the World Intellectual Property Organisation (WIPO), URL: https://www.wipo.int/classifications/ipc/en/.

[b] All outputs are defined according to Table 3.2. Breakdown of t4, t8, and t12 refers to the number of years of the granted patent after the filling date, and only exist if the patent has reached the respective age.

[c] The assessment of Tables 5.2 - 5.13 is synthesized by reviewing and ranking the performance of each model according to the following evaluation metrics in order (per output proxy and IPC classification): (i) the macro average F1-score for $\Theta \neq 0.50$, which is optimised to maximise the macro average F1-score (Lipton et al., 2014; Provost, 2000); (ii) the accuracy; and (iii) the false negative rate.

[d] Evaluation symbols: The numbers 1, 2, and 3 represent the ranking of IPC model performance per output proxy, + represents the lowest false negative model performance per output proxy for the IPC section, and x represents the highest ratio of V_H to V_L patents per output proxy for the IPC section.

As a rule-of-thumb, one could utilise the grant_lag model to forecast the value of patents in IPC E, the generality in IPC G, the quality_index_4, citations_t8, citations_t8, and renewal_t4 in IPC A, the citations_t4 and generality_t8 in IPC B, the generality_t4 in IPC C, the generality_t12 in IPC D, the renewal_t12 in IPC F, the generality in IPC G, and the renewal_t8 in IPC H. Moreover, in the short term (t4) time horizon, we observe that we can use citations_t4 and renewal_t4 together, with the weight in favour of citations_t4 due to the lower false negative rate. For the medium term (t8) time horizon, we can use citations_t8 and generality_t8, with the weight in favour of the generality_t8 due to the lower false negative rate and higher ratio of V_H to V_L patents. For the long term (t12) time horizon, citations_t12 and renewals_t12 can be used together.

[1] This is reflected in the models for which results are shown in Tables 5.2-5.13, which we trained, validated and tested per output proxy on the different IPC classifications.

We associate a different output proxy with different technology classes, for different time horizons. This has implications for firms by reducing time for technology development, as well as firms with limited resources, such as small-medium enterprises (SMEs). Firms can use Table 6.1 to tailor the time horizon and the technology they are developing, and use only the relevant proxies to forecast patent value. At the same time, SMEs with limited resources, can choose the most representative proxy for the technology they are developing to forecast an estimate of patent value, saving resources. Thus, we can use Table 6.1 to tailor the choice of output proxy models in the particular case for the time horizon and technology area.

6.2.3 Patent Text and Language

A methodological contribution of this research relates to the use of patent text for the valuation of patents (see 3.2.3). This seems to be lacking in the studies that are similar with this research (see Table 2.12). Some recent studies used text, however, mainly to classify patents in the respective technological areas. Lee & Hsiang (2019a) use the BERT transformer model to classify patents in the respective IPC sections using a pre-trained 12-layer deep neural network with 110 million parameters and 768 nodes per hidden layer. While our models are not structured in the same way to measure the same output, i.e. the probability of classifying a patent in the respected field, we decode the patent language with less complexity developing a deep and wide 7-layer and 2048 nodes per hidden layer neural network. The reduction in depth (i.e. the number of layers) together with the increase in the width (i.e. number of nodes in hidden layers), increases the model performance, indicating that the approach utilised has the advantage of a good representation of the patent text.

The use of both the metadata on patents and the text, in the form of a Doc2Vec embedding, seems to improve the overall classification (Choi et al., 2019; Li et al., 2018), in comparison to prior studies which use only numeric data such as Lee et al. (2018) and Noh & Lee (2020). A similar approach proposed by Lu et al. (2020) uses the patent text to predict the forward citation similarity between patents and arrives at comparable results to our proposed method. They use an adapted Doc2Vec methodology with the gensim library and a multi-layer perceptron (MLP) to find accuracies of 0.61 to 0.94 for similar patents. This is similar to the study by Abdelgawad et al. (2020) that use artificial neural networks (ANN) to analyse patent text for feature extraction and technology area classification. Our results seem to outperform prior studies in both accuracy and macro average F1-score. This is mainly due to using a deeper and wider network because of the language complexity found within patent documents (Hu et al., 2018a), as well as the development of the Doc2Vec methodology for every patent text section (abstract, claims, description, title) instead of using the whole document as a single vector leading to noise (Helmers et al., 2019).

Chapter 7

Conclusion

In this research, we develop an artificial intelligence (AI) deep learning approach for the valuation of patents to identify valuable patents. Unlike previous studies (chapter 2) that focus mostly on regression methods, small datasets (Table 2.7), and low capacity shallow artificial neural networks (Table 2.12), we propose an AI deep learning approach to predict the value of patents with large datasets and a variety of patent value proxies associated to patent value dimensions.

We develop a large USPTO dataset consisting of all granted patents from 1976-2019. We transform all patent data into: (i) numeric features, (ii) categoric features using one hot encoding (OHE), and (iii) text features using a Doc2Vec methodology for the patent text (abstract, claims, summary, title) (chapter 3). We then develop our deep learning approach, using deep and wide feed-forward artificial neural networks (ANN). We operationalise patent value determinants to predict 12 patent value proxies. We train, validate and test our deep learning models using our developed large dataset, to predict the grant_lag, generality, quality_index_4, citations_t4, citations_t8, citations_t12, generality_t4, generality_t8, generality_t12, renewal_t4, renewal_t8 and renewal_t12. We evaluate our models using 8 evaluation metrics, namely accuracy, confusion matrix, F1-score, false negative rate (FNR), log loss, mean absolute error (MAE), precision, recall, and different evaluation strategies to ensure the stability and generalising ability of our developed approach (chapter 4).

Our results show that our models have higher accuracy and macro average F1-scores. With increasing prediction horizons, we observe an increase in the macro average F1-scores. In addition, we find that the composite index that takes into consideration more than one value dimension, has the combined highest accuracy and macro average F1-score. Our study has moved away from relatively small datasets, limited to specific technology field, and allowed for reproducibility in other fields. We can tailor models to different technology area, with different patent value proxies, with different time horizons (chapter 5).

7.1 Addressing industry-related problems and implications

This research addresses several of the problem areas identified and explored by the technology roadmap approach (see 1.1.2). In chapter 3, we extract, clean and prepare the full USPTO dataset of granted patents, using state-of-the-art data preparation methods and advanced data management processes (see 3.1). In addition, we develop the Doc2Vec methodology to represent the patent text using natural language processing (NLP) methods (see 3.2.3). We also interconnect numerical, categorical, and text data from patent data (see 3.2). In addition, we develop a deep learning methodology with deep and wide artificial neural networks using multiple patent indicators (see chapter 4), with a transparent evaluation and reporting structure (see chapter 5).

This research belongs to the emerging field of IPI research and proposes an AI deep learning approach to predict the value of patents with academic and industrial implications. Since we use all USPTO granted patents from 1976-2019 to train our models, we can apply this approach to patents in any technology field. In particular, most previous studies have deployed lagging proxies, meaning that newly issued patents cannot be effectively valued. By contrast, our approach outperforms previous research (Table 2.12), addresses the industrial gap identified by the PADT roadmap (see 1.1.2) and enables researchers to value patents using a variety of patent value proxies, based on different value dimensions, tailored to specific technology areas.

Moreover, we could develop our approach in an automated intelligent system (Aristodemou & Tietze, 2017b), which predicts the likelihood that a patent is valuable V_H or not valuable V_L, by reducing the time and cost of a manual human expert patent valuation. This could be in the form of a decision support tool to effectively support experts (technology and innovation managers, IP managers etc.) in their decision making by providing data-driven *intellectual property intelligence (IPI)* from large amount of patent data. Experts could tailor the time horizon and technology fields with only the relevant proxies, to reduce time for technology development, and also identify key applicants or inventors of valuable patents in different technology fields. In addition, the trained models can be used to identify emerging technologies, monitor technological trends, and provide competitive technology intelligence. The long term (t12) time horizon models could identify granted patents or possible patent applications that could become valuable in the future, and thus identifying early stage valuable technologies. Firms with limited resources, such as small-medium enterprises (SMEs) can choose a representative proxy to forecast an estimate patent value and save resources. In addition, patent offices could use adapted versions of our proposed approach to identify invalid patents based on low value and their similarity to other patents or could use elements of the research, such as the Doc2Vec method and vectors, to expand their prior-art similarity

search for aiding the patent examination process. Consequently, the proposed approach could efficiently support experts in their judgement of patent value, policy making in the government's investments in technological sectors of the future to support the economy, and patent offices in their administrative work. Furthermore, linking patent value with products could make it possible to identify key patents for licensing for specific products, the technical feasibility and completeness of products, and the market value of products (chapter 6).

7.2 Limitations and future research

Despite this research contributions (see 1.3), there are limitations and challenges to overcome, which can form the basis of future research. We are witnessing a shift from traditional regression models to AI-based intellectual property intelligence (IPI) data-driven models, which can analyse big patent data. Given that we provide a first definition of the emerging IPI research stream, this has the potential to change how research is done, with researchers in this field (and subsequent data-driven fields) needing to acquire new competences and skills to effectively and collectively understand AI research.

Firstly, our results show a significant improvement from prior art, with higher accuracies and macro average F1-scores. The AI model's prediction performance could be improved further to forecast patent value proxies, especially in increasing the F1-score for forward citations. This can be explored in future research by: (i) using a variety of artificial intelligence (AI) algorithms, such as XGBoost, to report a comparison of approaches; (ii) improving the distinction between V_H and V_L patents further by improving the data pre-processing[1]; (iii) expanding on approaches such as the ensemble methodology of combining several AI methods (bagging, boosting, stacking) for prediction and then using a concatenation methods (such averaging) to combine the predictions from several approaches.

Secondly, our study focuses on exploring a wide range of patent value proxies and patent value determinants, linked to patent value dimensions. One of the limitations is the narrow focus on using only proxies arising from the patent data, with limited focus on the economic value of patents. The PatVal study has estimates of patent value (Gambardella et al., 2005). However, given the difficulty in sourcing this dataset, we focus on alternative proxies capable of estimating patent value, of which some correlate with the patent premium (Arora & Gambardella, 2010; Jensen et al., 2011). Future research could expand our approach to:

[1]Unfortunately, due to the computational resources required to run these sort of algorithms, it was not possible to train, validate and test all output categorical proxies on the full dataset. This is partly the reason we have developed the wide evaluation strategy in 4.5. Given the emerging field of IPI research, the decreasing costs of computational resources, and the availability of resources and cloud services to a wider audience, future researchers could also attempt to train AI models on the full dataset (100FD).

(i) include other datasources such as financial data, social opinion data (Twitter, Facebook and social media sites) and product related data sources; (ii) develop the ensemble method approach to include a reinforcement learning element, where the human experts' estimates of the value of patents act as feedback response (reward) to improve the learning of the AI models to identify valuable patents. The later has the potential to develop into an intelligent system that can collectively take into consideration all types of learning, such as supervised, unsupervised, and reinforcement, to arrive at a rational decision that maximises the cumulative reward. In addition, future research could focus on the strategic and social (sustainable, green etc.) value dimensions and the identification/development of proxies and determinants. This can have an important impact on the patent value literature with comparative studies and construction of composite indices.

Moreover, we develop an AI deep learning approach, which is based on deep and wide 7-layer artificial neural networks, and are widely applicable. We have utilised numeric, categoric and text features that have had a significant increase in the AI models prediction performance. We have developed a Doc2Vec methodology to represent the patent text into vector space embeddings. Future research can focus on different AI text models, with a focus on: (i) the contribution of the individual text sections to patent value (abstract, claims - independent and dependent, summary, title), and (ii) the contributions of different natural language processing (NLP) models in understanding the syntactic and semantic elements of the text. Researchers can focus on the following NLP models: (i) developing joint Word2Vec, Paragraph2Vec, Doc2Vec vector space models (VSM), with the proximity in the syntactic and semantic elements of the text, and clustering similar patents together; (ii) with extensions of recurrent neural network (RNN) models that learn sequential, syntactic and semantic elements of the text with the aim to reproduce the text from their learnings. This is an emerging field that has recently spark a large interest in the AI NLP community of using the BERT transformer model. This has the potential to significantly impact the work of patent offices[1] by (i) identifying invalid patents when considering the filing date, scope, inventive nature and novelty of patented inventions, and (ii) improving the information retrieval of similar patents during the prior-art search in the patent examination[2].

[1]These type of AI models could also have a significant impact on the overall question "Can AI be an inventor?", since the recurrent neural network (RNN) approach has the potential to re-create text, and thus has inherited ability to produce novel text taking into consideration the elements learned during training.

[2]The richness of information found within patent data, makes an attractive source of information. This is because it combines numeric, categoric and text elements, which several proxies, determinants and indicators can be constructed, for economic and strategy research. In particular, the text found within patents has the potential: (i) to describe products and technologies; (ii) to unveil legal dynamics found within the language that describe patented inventions; and (iii) to identify innovation diffusion dynamics with the introduction of complementary and substitute products, and how these with the strategic patenting activity of firms represented by the legal language.

In addition, our approach has moved away from relatively small datasets, limited to a specific technological field. From Table 6.1, we can associate a different output proxy to a different technology class for different time horizons. Future research can focus on creating industry-related validated models to be used by small firms, SMEs and large firms, depending on the different needs. As part of the Strategic Technology and Innovation Management (STIM) consortium, we explore the applications of the AI patent value models in specific tasks of innovation management. Further research includes the industrial applicability and thus building a number of case study validation protocols with firms from a variety of sectors, to identify valuable patents, and perform some portfolio analysis (patent level, firm level, technology level, industry level). We anticipate this research would be interesting for future researchers working in the area of reinforcement learning where the feedback from the environment (in this case the firm experts), can be taken as a signal input to the model to improve its performance.

Future researchers could explore further AI methods for big patent data using deep learning and moving towards the direction of reinforcement learning, which include: (i) the disambiguation of patent applicants and names; (ii) the interconnectedness of patent data to other data sources; (iii) the identification of invalid patents, (iv) the identification of prior-art; (v) improvements in patent translation and harmonisation of data from patent offices; (vi) virtual reality interfaces with patent-to-product interaction; (vii) automating patent drafting; (viii) AI inventing; (ix) identification of opposition and litigation cases; (x) transparent identification of licensing agreements; and (xi) AI examination of filed patents etc. A further application could focus on using AI patent value models together with blockchain technologies that track licensing agreements, to predict future revenues or the asset value of patents and intangibles value. One could take this even further to build prototypes of neural network models that could negotiate contracts themselves based on the value of the patented inventions, the innovation landscape, the micro and macro economic environments. With AI and big data and the ability to process large amounts of information, this has the potential to impact accounting standards and quantify the value of intangible assets based on a negotiating contract price, licensing agreements, and the collective information analysis about the value of the patented invention from other datasources.

References

Abadi, M., et al. (2016a). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *Google Research working paper series* http://arxiv.org/abs/1603.04467.

Abadi, M., et al. (2016b). TensorFlow: A system for Large-Scale Machine Learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)* Savannah, GA, USA https://tensorflow.org.

Abbas, A., Zhang, L., & Khan, S. U. (2014). A literature review on the state-of-the-art in patent analysis. *World Patent Information*, *37*, 3–13, https://doi.org/10.1016/j.wpi.2013.12.006.

Abbott, R. (2018). Everything Is Obvious. *U.C.L.A. Law Review*, *2*, 1–51 https://ssrn.com/abstract=3056915.

Abdelgawad, L., Kluegl, P., Genc, E., Falkner, S., & Hutter, F. (2020). Optimizing Neural Networks for Patent Classification. In U. Brefeld, E. Fromont, A. Hotho, A. Knobbe, M. Maathuis, & C. Robardet (Eds.), *Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2019. Lecture Notes in Computer Science* (pp. 688–703). Springer International Publishing.

Abood, A. & Feltenberger, D. (2018). Automated patent landscaping. *Artificial Intelligence and Law*, *26*(2), 103–125, https://doi.org/10.1007/s10506-018-9222-4.

Adams, S. R. (2006). *Information Sources in Patents*. Gale virtual reference library. De Gruyter https://books.google.co.uk/books?id=zQR197ztoBEC.

Agarwal, S. & Yu, H. (2009). Automatically classifying sentences in full-text biomedical articles into introduction, methods, results and discussion. *Bioinformatics*, *25*(23), 3174–3180, https://doi.org/10.1093/bioinformatics/btp548.

Agatonovic-Kustrin, S. & Beresford, R. (2000). Basic concepts of artificial neural network (ANN) modelling and its application in pharmaceutical research. *Journal of Pharmaceutical and Biomedical Analysis*, *22*(5), https://doi.org/10.1016/S0731-7085(99)00272-1.

Agrawal, A., Gans, J., & Goldfarb, A. (2017). How AI Will Change the Way We Make Decisions. *Harvard Business Review*, *July*, 1–7 https://hbr.org/2017/07/how-ai-will-change-the-way-we-make-decisions.

Agresti, A. (2015). *Foundations of Linear and Generalized Linear Models*. Wiley Series in Probability and Statistics, John Wiley & Sons https://books.google.com.cy/books?id=jlIqBgAAQBAJ.

Akhondi, S. A., et al. (2016). Chemical entity recognition in patents by combining dictionary-based and statistical approaches. *Database: the journal of biological databases and curation*, *2016*(December 2017), 1–8, https://doi.org/10.1093/database/baw061.

Albert, M., Avery, D., Narin, F., & McAllister, P. (1991). Direct validation of citation counts as indicators of industrially important patents. *Research Policy*, *20*(3), 251–259, https://doi.org/10.1016/0048-7333(91)90055-U.

Alcácer, J. & Gittelman, M. (2006). Patent Citations as a Measure of Knowledge Flows: The Influence of Examiner Citations. *Review of Economics and Statistics*, *88*(4), 774–779, https://doi.org/10.1162/rest.88.4.774.

Alcácer, J., Gittelman, M., & Sampat, B. (2009). Applicant and examiner citations in U.S. patents: An overview and analysis. *Research Policy*, *38*(2), 415–427, https://doi.org/10.1016/j.respol.2008.12.001.

Alcácer, V. & Cruz-Machado, V. (2019). Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems. *Engineering Science and Technology, an International Journal*, *22*(3), 899–919, https://doi.org/10.1016/j.jestch.2019.01.006.

Altini, M. (2015). Dealing with imbalanced data: undersampling, oversampling and proper cross-validation. *marcoaltini.com blog* https://www.marcoaltini.com/blog/dealing-with-imbalanced-data-undersampling-oversampling-and-proper-cross-validation.

Altuntas, S., Dereli, T., & Kusiak, A. (2015). Forecasting technology success based on patent data. *Technological Forecasting and Social Change*, *96*, 202–214, https://doi.org/10.1016/j.techfore.2015.03.011.

Aras, H., Türker, R., Geiss, D., Milbradt, M., & Sack, H. (2018). Get your hands dirty: Evaluating Word2Vec models for patent data. In *CEUR Workshop Proceedings*, volume 2198 (pp. 2–5).

Aristodemou, L. (2020a). PhD Code Submission on Github, https://github.com/LAristodemou/PhD_Submission_Code/.

Aristodemou, L. (2020b). PhD Dataset Submission on Google Drive File Stream.

Aristodemou, L. & Tietze, F. (2017a). A literature review on the state-of-the art on Intellectual Property Analytics (IPA). *Centre for Technology Management working paper series*, *2*(Nov), 1–14, https://doi.org/10.17863/CAM.13928.

Aristodemou, L. & Tietze, F. (2017b). *Exploring the Future of Patent Analytics*. Technical report, Centre for Technology Management, Institute for Manufacturing, University of Cambridge, Cambridge, UK.

Aristodemou, L. & Tietze, F. (2018a). Citations as a measure of technological impact: A review of forward citation-based measures. *World Patent Information*, *53*(April 2017), 39–44, https://doi.org/10.1016/j.wpi.2018.05.001.

Aristodemou, L. & Tietze, F. (2018b). The state-of-the-art on Intellectual Property Analytics (IPA): A literature review on artificial intelligence, machine learning and deep learning methods for analysing intellectual property (IP) data. *World Patent Information*, *55*(April), 37–51, https://doi.org/10.1016/j.wpi.2018.07.002.

Aristodemou, L. & Tietze, F. (2019a). Early Stage Identification of Valuable Technologies: a Deep Learning approach. In *European Policy for Intellectual Property (EPIP) Conference 2019, Zurich, Switzerland*.

Aristodemou, L. & Tietze, F. (2019b). Technology Strategic Decision Making (SDM): an overview of decision theories, processes and methods. *Centre for Technology Management working paper series*, *5*(Jan), 1–21, https://doi.org/https://doi:10.17863/CAM.35691.

Aristodemou, L. & Tietze, F. (2020). Technological value characteristics and identification of valuable technologies: a novel deep learning based methodology using patent data and intellectual property analytics. In *R&D Management Conference 2020, University of Strathclyde, Glasgow, Scotland, United Kingdom* (pp. Conference postponed due to COVID–19).

Aristodemou, L., Tietze, F., Athanassopoulou, N., & Minshall, T. (2017a). Exploring the Future of Patent Analytics: A Technology Roadmapping Approach. In *R&D Management Conference 2017, Leuven, Belgium*, volume November (pp. 1–9).

Aristodemou, L., Tietze, F., Athanassopoulou, N., & Minshall, T. (2017b). Exploring the Future of Patent Analytics A Technology Roadmapping approach. *Centre for Technology Management working paper series*, *5*(Nov), 1–10, https://doi.org/10.17863/CAM.13967.

Aristodemou, L., Tietze, F., Brintrup, A., & Deeble, S. (2018). Early Stage Technology Strategic Decision Making : a machine learning approach using Intellectual Property Analytics. In *R&D Management Conference 2018, Milan, Italy* (pp. 1–9).

Aristodemou, L., Tietze, F., Brintrup, A., & Deeble, S. (2019a). Intellectual Property Analytics Decision Support Tool (IPDST) for Early Stage Technology Decision Making. *Centre for Technology Management working paper series*, *1*(Jan), 1–6, https://doi.org/10.17863/CAM.35544.

Aristodemou, L., Tietze, F., O'Leary, E., & Shaw, M. (2019b). A Literature Review on Technology Development Process (TDP) Models. *Centre for Technology Management working paper series*, *6*(Jan), 1–31, https://doi.org/https://doi:10.17863/CAM.35692.

Aristodemou, L., Tietze, F., & Shaw, M. (2020). Stage Gate Decision making: a scoping review of Technology Strategic Selection Criteria for Early Stage Projects. *IEEE Engineering Management Review*, *8581*(c), https://doi.org/10.1109/EMR.2020.2985040.

Arora, A., Ceccagnoli, M., & Cohen, W. M. (2008). R&D and the patent premium. *International Journal of Industrial Organization*, *26*(5), 1153–1179, https://doi.org/10.1016/j.ijindorg.2007.11.004.

Arora, A. & Gambardella, A. (2010). Chapter 15 - The Market for Technology. In B. H. Hall & N. Rosenberg (Eds.), *Handbook of The Economics of Innovation, Vol. 1*, volume 1 of *Handbook of the Economics of Innovation* (pp. 641 – 678). North-Holland http://www.sciencedirect.com/science/article/pii/S0169721810010154.

Arora, S., Hu, W., & Kothari, P. K. (2018). An Analysis of the t-SNE Algorithm for Data Visualization. In *Conference on Learning Theory (COLT)* http://arxiv.org/abs/1803.01768.

Arrow, K. J. & Intriligator, M. D. (2010). Introduction to the Series. In B. H. Hall & N. Rosenberg (Eds.), *Handbook of The Economics of Innovation, Vol. 1*, volume 1 of *Handbook of the Economics of Innovation*. North-Holland http://www.sciencedirect.com/science/article/pii/S016972181001018X.

Arts, S., Appio, F. P., & Van Looy, B. (2013). Inventions shaping technological trajectories: do existing patent indicators provide a comprehensive picture? *Scientometrics*, *97*(2), 397–419, https://doi.org/10.1007/s11192-013-1045-1.

Baglieri, D. & Cesaroni, F. (2013). Capturing the real value of patent analysis for R&D strategies. *Technology Analysis & Strategic Management*, *25*(8), 971–986, https://doi.org/10.1080/09537325.2013.823149.

Bakker, J. (2017). The log-linear relation between patent citations and patent value. *Scientometrics*, *110*(2), 879–892, https://doi.org/10.1007/s11192-016-2208-7.

Barredo Arrieta, A., et al. (2020). Explainable Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, *58*, 82–115, https://doi.org/10.1016/j.inffus.2019.12.012.

Baruffaldi, S., Van Beuzekom, B., Dernis, H., Harhoff, D., Rao, N., Rosenfeld, D., & Squicciarini, M. (2020). Identifying and measuring developments in artificial intelligence : Making the impossible possible. *OECD Science, Technology and Industry Working Papers*, (pp. 1–68)., https://doi.org/https://dx.doi.org/10.1787/5f65ff7e-en.

Basheer, I. & Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, *43*, 3–31, https://doi.org/10.1016/S0167-7012(00)00201-3.

Bass, S. D. & Kurgan, L. A. (2010). Discovery of factors influencing patent value based on machine learning in patents in the field of nanotechnology. *Scientometrics*, *82*(2), 217–241, https://doi.org/10.1007/s11192-009-0008-z.

Baudour, F. & van de Kuilen, A. (2015). Evolution of the Patent Information World - Challenges of yesterday, today and tomorrow. *World Patent Information*, *40*, 4–9, https://doi.org/10.1016/j.wpi.2014.10.001.

Becker, W., Worden, K., & Rowson, J. (2013). Bayesian sensitivity analysis of bifurcating nonlinear models. *Mechanical Systems and Signal Processing*, *34*(1-2), 57–75, https://doi.org/10.1016/j.ymssp.2012.05.010.

Bekkers, R., Bongard, R., & Nuvolari, A. (2011). An empirical study on the determinants of essential patent claims in compatibility standards. *Research Policy*, *40*(7), 1001–1015, https://doi.org/10.1016/j.respol.2011.05.004.

Beleites, C., Neugebauer, U., Bocklitz, T., Krafft, C., & Popp, J. (2013). Sample size planning for classification models. *Analytica Chimica Acta*, *760*(June 2012), 25–33, https://doi.org/10.1016/j.aca.2012.11.007.

Belkina, A. C., Ciccolella, C. O., Anno, R., Halpert, R., Spidlen, J., & Snyder-Cappione, J. E. (2019). Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nature Communications*, *10*(1), 1–12, https://doi.org/10.1038/s41467-019-13055-y..

Benaroch, M. (2001). Option-based management of technology investment risk. *IEEE Transactions on Engineering Management*, *48*(4), 428–444, https://doi.org/10.1109/17.969422.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7700 LECTU*, 437–478, https://doi.org/10.1007/978-3-642-35289-8-26.

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy Layer-Wise Training of Deep Networks. *Advances in Neural Information Processing Systems*, *19*(1), 153, https://doi.org/citeulike-article-id:4640046 http://papers.nips.cc/paper/3048-greedy-layer-wise-training-of-deep-networks.pdf.

Benvenuto, N. & Piazza, F. (1992). On the Complex Backpropagation Algorithm. *IEEE Transactions on Signal Processing*, *40*(4), 967–969, https://doi.org/10.1109/78.127967.

Berg, S., Wustmans, M., & Bröring, S. (2018). Identifying first signals of emerging dominance in a technological innovation system: A novel approach based on patents. *Technological Forecasting and Social Change*, *146*(January 2018), 706–722, https://doi.org/10.1016/j.techfore.2018.07.046.

Bergdahl, M., et al. (2007). *Handbook on Data Quality Assessment Methods and Tools*. Technical report, European Commission, Eurostat.

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011* (pp. 1–9).

Bergstra, J., Yamins, D., & Cox, D. D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. *30th International Conference on Machine Learning, ICML 2013*, (PART 1), 115–123.

Bessen, J. (2008). The value of U.S. patents by owner and patent characteristics. *Research Policy*, *37*(5), 932–945, https://doi.org/10.1016/j.respol.2008.02.005.

Bessen, J. (2009). Estimates of patent rents from firm market value. *Research Policy*, *38*(10), 1604–1616, https://doi.org/10.1016/j.respol.2009.09.014.

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media Inc., annotated edition http://nltk.org/book.

Bishop, C. M. (2006). *Pattern Recogniton and Machine Learning*. Information Science and Statistics, Springer (India) Private Limited, 2nd edition.

Bjorck, J., Gomes, C., Selman, B., & Weinberger, K. Q. (2018). Understanding batch normalization. *Advances in Neural Information Processing Systems*, *2018-Decem*(NeurIPS), 7694–7705.

Blind, K., Cremers, K., & Mueller, E. (2009). The influence of strategic patenting on companies' patent portfolios. *Research Policy*, *38*(2), 428–436, https://doi.org/10.1016/j.respol.2008.12.003.

Blind, K., Edler, J., Frietsch, R., & Schmoch, U. (2006). Motives to patent: Empirical evidence from Germany. *Research Policy*, *35*(5), 655–672, https://doi.org/10.1016/j.respol.2006.03.002.

Bond, S. & Meghir, C. (1994). Dynamic investment models and the firm's financial policy. *Review of Economic Studies*, *61*(2), 197–222, https://doi.org/10.2307/2297978.

Bonino, D., Ciaramella, A., & Corno, F. (2010). Review of the state-of-the-art in patent information and forthcoming evolutions in intelligent patent informatics. *World Patent Information*, *32*(1), 30–38, https://doi.org/10.1016/j.wpi.2009.05.008.

Breitzman, a. F. & Mogee, M. E. (2002). The many applications of patent analysis. *Journal of Information Science*, *28*(3), 187–205, https://doi.org/10.1177/016555150202800302.

Bresnahan, T. (2010). Chapter 18 - General Purpose Technologies. In B. H. Hall & N. Rosenberg (Eds.), *Handbook of the Economics of Innovation, Volume 2*, volume 2 of *Handbook of the Economics of Innovation* (pp. 761 – 791). North-Holland http://www.sciencedirect.com/science/article/pii/S0169721810020022.

Bringsjord, S. & Govindarajulu, N. S. (2020). *Artificial Intelligence*. The Stanford Encyclopedia of Philosophy, Metaphysics Research Lab, Stanford University https://plato.stanford.edu/cgi-bin/encyclopedia/archinfo.cgi?entry=artificial-intelligence.

Bronshtein, A. (2017). Train/Test Split and Cross Validation in Python. *Towards Data Science* https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6.

Brown, S. J., Goetzmann, W., Ibbotson, R. G., & Ross, S. A. (1992). Survivorship Bias in Performance Studies. *The Review of Financial Studies*, *5*(4), 553–580 http://www.jstor.org/stable/2962141.

Brownlee, J. (2014). Discover Feature Engineering, How to Engineer Features and How to Get Good at It. *Machine Learning Mastery* https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/.

Brownlee, J. (2016a). Dropout Regularization in Deep Learning Models With Keras. *Machine Learning Mastery* https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/.

Brownlee, J. (2016b). How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras. *Machine Learning Mastery* https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models-python-keras/.

Brownlee, J. (2016c). Using Learning Rate Schedules for Deep Learning Models in Python with Keras. *Machine Learning Mastery* https://machinelearningmastery.com/using-learning-rate-schedules-deep-learning-models-python-keras/.

Brownlee, J. (2017a). A Gentle Introduction to Transfer Learning for Deep Learning. *Machine Learning Mastery* https://machinelearningmastery.com/transfer-learning-for-deep-learning/.

Brownlee, J. (2017b). Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. *Machine Learning Mastery* https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.

Brownlee, J. (2017c). How to Clean Text for Machine Learning with Python. *Machine Learning Mastery* https://machinelearningmastery.com/clean-text-machine-learning-python/.

Brownlee, J. (2017d). How to Develop Word Embeddings in Python with Gensim. *Machine Learning Mastery*.

Brownlee, J. (2017e). How to Evaluate the Skill of Deep Learning Models. *Machine Learning Mastery* https://machinelearningmastery.com/evaluate-skill-deep-learning-models/.

Brownlee, J. (2017f). How to One Hot Encode Sequence Data in Python. *Machine Learning Mastery* https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/.

Brownlee, J. (2017g). What is the Difference Between Test and Validation Datasets? *Machine Learning Mastery* https://machinelearningmastery.com/difference-test-validation-datasets/.

Brownlee, J. (2017h). Why One-Hot Encode Data in Machine Learning? *Machine Learning Mastery* https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/.

Brownlee, J. (2018a). A Gentle Introduction to Dropout for Regularizing Deep Neural Networks. *Machine Learning Mastery* https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/.

Brownlee, J. (2018b). A Gentle Introduction to k-fold Cross-Validation. *Machine Learning Mastery* https://machinelearningmastery.com/k-fold-cross-validation/.

Brownlee, J. (2018c). A Gentle Introduction to the Bootstrap Method. *Machine Learning Mastery* https://machinelearningmastery.com/a-gentle-introduction-to-the-bootstrap-method/.

Brownlee, J. (2018d). How to Avoid Overfitting in Deep Learning Neural Networks. *Machine Learning Mastery* https://machinelearningmastery.com/introduction-to-regularization-to-reduce-overfitting-and-improve-generalization-error/.

Brownlee, J. (2018e). How to Create a Bagging Ensemble of Deep Learning Models in Keras. *Machine Learning Mastery* https://machinelearningmastery.com/how-to-create-a-random-split-cross-validation-and-bagging-ensemble-for-deep-learning-in-keras/.

Brownlee, J. (2018f). How to Use Weight Decay to Reduce Overfitting of Neural Network in Keras. *Machine Learning Mastery* https://machinelearningmastery.com/how-to-reduce-overfitting-in-deep-learning-with-weight-regularization/.

Brownlee, J. (2018g). Use Weight Regularization to Reduce Overfitting of Deep Learning Models. *Machine Learning Mastery* https://machinelearningmastery.com/weight-regularization-to-reduce-overfitting-of-deep-learning-models/.

Brownlee, J. (2019a). 14 Different Types of Learning in Machine Learning. *Machine Learning Mastery* https://machinelearningmastery.com/types-of-learning-in-machine-learning/.

Brownlee, J. (2019b). 8 Tricks for Configuring Backpropagation to Train Better Neural Networks. *Machine Learning Mastery* https://machinelearningmastery.com/best-advice-for-configuring-backpropagation-for-deep-learning-neural-networks/.

Brownlee, J. (2019c). A Gentle Introduction to Batch Normalization for Deep Neural Networks. *Machine Learning Mastery* https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/.

Brownlee, J. (2019d). A Gentle Introduction to the Rectified Linear Unit (ReLU). *Machine Learning Mastery* https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/.

Brownlee, J. (2019e). Framework for Better Deep Learning. *Machine Learning Mastery* https://machinelearningmastery.com/framework-for-better-deep-learning/.

Brownlee, J. (2019f). How to Accelerate Learning of Deep Neural Networks With Batch Normalization. *Machine Learning Mastery* https://machinelearningmastery.com/how-to-accelerate-learning-of-deep-neural-networks-with-batch-normalization/.

Brownlee, J. (2019g). How to Control Neural Network Model Capacity With Nodes and Layers. *Machine Learning Mastery* https://machinelearningmastery.com/how-to-control-neural-network-model-capacity-with-nodes-and-layers/.

Brownlee, J. (2019h). How to use Learning Curves to Diagnose Machine Learning Model Performance. *Machine Learning Mastery* https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/.

Brownlee, J. (2019i). Loss and Loss Functions for Training Deep Learning Neural Networks. *Machine Learning Mastery* https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/.

Brownlee, J. (2019j). Neural Networks: Tricks of the Trade Review. *Machine Learning Mastery* https://machinelearningmastery.com/neural-networks-tricks-of-the-trade-review/.

Brownlee, J. (2019k). Recommendations for Deep Learning Neural Network Practitioners. *Machine Learning Mastery* https://machinelearningmastery.com/recommendations-for-deep-learning-neural-network-practitioners/.

Brownlee, J. (2019l). Tune Hyperparameters for Classification Machine Learning Algorithms. *Machine Learning Mastery* https://machinelearningmastery.com/hyperparameters-for-classification-machine-learning-algorithms/.

Brownlee, J. (2019m). What is Deep Learning? *Machine Learning Mastery* https://machinelearningmastery.com/what-is-deep-learning/.

Brownlee, J. (2020a). A Gentle Introduction to the Fbeta-Measure for Machine Learning. *Machine Learning Mastery - Imbalanced Classification* https://machinelearningmastery.com/fbeta-measure-for-machine-learning/.

Brownlee, J. (2020b). A Gentle Introduction to Threshold-Moving for Imbalanced Classification. *Machine Learning Mastery* https://machinelearningmastery.com/threshold-moving-for-imbalanced-classification/.

Brownlee, J. (2020c). Failure of Classification Accuracy for Imbalanced Class Distributions. *Machine Learning Mastery* https://machinelearningmastery.com/failure-of-accuracy-for-imbalanced-class-distributions/.

Brownlee, J. (2020d). How to Use One-vs-Rest and One-vs-One for Multi-Class Classification. *Machine Learning Mastery* https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/.

Brownlee, J. (2020e). Neural Networks are Function Approximation Algorithms. *Machine Learning Mastery* https://machinelearningmastery.com/neural-networks-are-function-approximators/.

Brownlee, J. (2020f). Tour of Evaluation Metrics for Imbalanced Classification. *Machine Learning Mastery* https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/.

Brownlee, J. (2020g). Train-Test Split for Evaluating Machine Learning Algorithms. *Machine Learning Mastery* https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/.

Brownlee, J. (2020h). Understand the Impact of Learning Rate on Neural Network Performance. *Machine Learning Mastery* https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/.

Budhiraja, A. (2016). Dropout in (Deep) Machine learning https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5.

Bushaev, V. (2018). Adam — latest trends in deep learning optimization. https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c.

Cai, L. & Zhu, Y. (2015). The Challenges of Data Quality and Data Quality Assessment in the Big Data Era. *Data Science Journal*, *14*(0), 2, https://doi.org/10.5334/dsj-2015-002 http://datascience.codata.org/article/10.5334/dsj-2015-002/.

Callaert, J., Grouwels, J., & van Looy, B. (2012). Delineating the scientific footprint in technology: Identifying scientific publications within non-patent references. *Scientometrics*, *91*(2), 383–398, https://doi.org/10.1007/s11192-011-0573-9.

Cao, Y. & Wang, L. (2017). Automatic Selection of t-SNE Perplexity. In *ICML 2017 AutoML Workshop (JMLR: Workshop and Conference Proceedings 1)* (pp. 1–7). http://arxiv.org/abs/1708.03229.

Caruana, R., Lawrence, S., & Giles, L. (2001). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in Neural Information Processing Systems*.

Carvalho, D. S. & Nguyen, M. L. (2017). Efficient neural-based patent document segmentation with term order probabilities. In *ESANN 2017 - Proceedings, 25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, number 4 (pp. 171–176).

Chandra, A. L. (2018). Perceptron Learning Algorithm: A Graphical Explanation Of Why It Works https://towardsdatascience.com/perceptron-learning-algorithm-d5db0deab975.

Chandra, P. & Dong, A. (2018). The relation between knowledge accumulation and technical value in interdisciplinary technologies. *Technological Forecasting and Social Change*, *128*(December 2017), 235–244, https://doi.org/10.1016/j.techfore.2017.12.006.

Chang, S.-B. B. (2012). Using patent analysis to establish technological position: Two different strategic approaches. *Technological Forecasting and Social Change*, *79*(1), 3–15, https://doi.org/10.1016/j.techfore.2011.07.002.

Chen, D. Z., Lin, W. Y. C., & Huang, M. H. (2007). Using Essential Patent Index and Essential Technological Strength to evaluate industrial technological innovation competitiveness. *Scientometrics*, *71*(1), 101–116, https://doi.org/10.1007/s11192-007-1655-6.

Chen, G., Chen, P., Shi, Y., Hsieh, C.-Y., Liao, B., & Zhang, S. (2019). Rethinking the Usage of Batch Normalization and Dropout in the Training of Deep Neural Networks http://arxiv.org/abs/1905.05928.

Chen, H., Engkvist, O., Wang, Y., Olivecrona, M., & Blaschke, T. (2018). The rise of deep learning in drug discovery. *Drug Discovery Today*, *23*(6), 1241–1250, https://doi.org/10.1016/j.drudis.2018.01.039.

Chen, Y. S. & Chang, K. C. (2009). Using neural network to analyze the influence of the patent performance upon the market value of the US pharmaceutical companies. *Scientometrics*, *80*(3), 637–655, https://doi.org/10.1007/s11192-009-2095-2.

Chen, Y. S. & Chang, K. C. (2010a). Analyzing the nonlinear effects of firm size, profitability, and employee productivity on patent citations of the US pharmaceutical companies by using artificial neural network. *Scientometrics*, *82*(1), 75–82, https://doi.org/10.1007/s11192-009-0034-x.

Chen, Y. S. & Chang, K. C. (2010b). Exploring the nonlinear effects of patent citations, patent share and relative patent position on market value in the US pharmaceutical industry. *Technology Analysis and Strategic Management*, *22*(2), 153–169, https://doi.org/10.1080/09537320903498496.

Chen, Y. S. & Chang, K. C. (2010c). The nonlinear nature of the relationships between the patent traits and corporate performance. *Scientometrics*, *82*(1), 201–210, https://doi.org/10.1007/s11192-009-0101-3.

Chen, Y. S. & Chang, K. C. (2010d). The relationship between a firm's patent quality and its market value - The case of US pharmaceutical industry. *Technological Forecasting and Social Change*, *77*(1), 20–33, https://doi.org/10.1016/j.techfore.2009.06.003.

Cheng, H.-T., et al. (2017). Deep learning for recommender systems. In *RecSys 2017 - Proceedings of the 11th ACM Conference on Recommender Systems* (pp. 396–397).: Google.

Chesbrough, H. W. (2003a). The era of open innovation. *MIT Sloan Management Review*.

Chesbrough, H. W. (2003b). *The new imperative for creating and profiting from technology*. Harvard Business Publishing.

Chiang, T.-A., Wu, C.-Y., Trappey, C., & Trappey, A. (2011). An intelligent system for automated binary knowledge document classification and content analysis. *Journal of Universal Computer Science*, *17*(14), 1991–2008.

Choi, J., Jang, D., Jun, S., & Park, S. (2015). A Predictive Model of Technology Transfer Using Patent Analysis. *Sustainability*, *7*(12), 16175–16195, https://doi.org/10.3390/su71215809.

Choi, J., Jeong, B., Yoon, J., Coh, B. Y., & Lee, J. M. (2020). A novel approach to evaluating the business potential of intellectual properties: A machine learning-based predictive analysis of patent lifetime. *Computers and Industrial Engineering*, *145*(May), 106544, https://doi.org/10.1016/j.cie.2020.106544.

Choi, S., Lee, H., Park, E. L., & Choi, S. (2019). Deep Patent Landscaping Model Using Transformer and Graph Embedding. *Working paper article* http://arxiv.org/abs/1903.05823.

Chollet, F. & Others (2015). Keras https://keras.io.

Cortes, C., Mohri, M., & Rostamizadeh, A. (2009). L2 Regularization for Learning Kernels. *Google Research working paper series (UAI2009)*.

Criscuolo, P. & Verspagen, B. (2008). Does it matter where patent citations come from? Inventor vs. examiner citations in European patents. *Research Policy*, *37*, 1892–1908, https://doi.org/10.1016/j.respol.2008.07.011.

Cronin, P., Ryan, F., & Coughlan, M. (2008). Undertaking a literature review : a step-by-step approach. *British Journal of Nursing*, *17*(1), 38–43.

Cross Validated (2010). How to choose the number of hidden layers and nodes in a feedforward neural network? https://stats.stackexchange.com/questions/181/how-to-choose-the-number-of-hidden-layers-and-nodes-in-a-feedforward-neural-netw.

Cui, Y., Jia, M., Lin, T. Y., Song, Y., & Belongie, S. (2019). Class-balanced loss based on effective number of samples. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2019-June*, 9260–9269, https://doi.org/10.1109/CVPR.2019.00949.

Dai, A. M., Olah, C., & Le, Q. V. (2015). Document Embedding with Paragraph Vectors. *Google Research working paper series*, (pp. 1–8). http://arxiv.org/abs/1507.07998.

Darken, C., Chang, J., & Moody, J. (1992). Learning rate schedules for faster Stochastic gradient search. In *Neural Networks for Signal Processing - Proceedings of the IEEE Workshop*.

De Clercq, D., Diop, N. F., Jain, D., Tan, B., & Wen, Z. (2019). Multi-label classification and interactive NLP-based visualization of electric vehicle patent data. *World Patent Information*, *58*(July), 101903, https://doi.org/10.1016/j.wpi.2019.101903.

De Fauw, J., et al. (2018). *Clinically applicable deep learning for diagnosis and referral in retinal disease*, volume 24. Springer US http://www.nature.com/articles/s41591-018-0107-6.

de la Paz-Marín, M., Campoy-Muñoz, P., & Hervás-Martínez, C. (2012). Non-linear multiclassifier model based on Artificial Intelligence to predict research and development performance in European countries. *Technological Forecasting and Social Change*, *79*(9), 1731–1745, https://doi.org/10.1016/j.techfore.2012.06.001.

De Saint-Georges, M. & Van Pottelsberghe De La Potterie, B. (2013). A quality index for patent systems. *Research Policy*, *42*(3), 704–719, https://doi.org/10.1016/j.respol.2012.09.003.

DeepAI (2020). Activation Function Definition https://deepai.org/machine-learning-glossary-and-terms/activation-function.

Delalleau, O. & Bengio, Y. (2011). Shallow vs. deep sum-product networks. *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011*, (pp. 1–9).

Deng, Y. (2007). Private value of European patents. *European Economic Review*, *51*(7), 1785–1812, https://doi.org/10.1016/j.euroecorev.2006.09.005.

Deng, Z., Lev, B., & Narin, F. (1999). Science and Technology as Predictors of Stock Performance. *Financial Analysts Journal*, *55*(3), 20–32, https://doi.org/10.2469/faj.v55.n3.2269.

Dercksen, K. (2018). Micro-averaged F1 optimization using neural networks https://koendercksen.com/micro-averaged-f1-optimization-using-neural-networks.html.

Dernis, H., Gkotsis, P., Grassano, N., Nakazato, S., Squicciarini, M., van Beuzekom, B., & Vezzani, A. (2019). *World Corporate Top R&D investors: Shaping the Future of Technologies and of AI*. Technical report, Publications Office of the European Union, Luxembourg.

Dintzner, J. P. & Van Thieleny, J. (1991). Image handling at the European Patent Office: BACON and first page. *World Patent Information*, *13*(3), 152–154, https://doi.org/10.1016/0172-2190(91)90070-L.

Dirnberger, D. (2011). A guide to efficient keyword, sequence and classification search strategies for biopharmaceutical drug-centric patent landscape searches - A human recombinant insulin patent landscape case study. *World Patent Information*, *33*(2), 128–143, https://doi.org/10.1016/j.wpi.2010.12.003.

Dobbin, K. K. & Simon, R. M. (2011). Optimally splitting cases for training and testing high dimensional classifiers. *BMC Medical Genomics*, *4*(1), 31, https://doi.org/10.1186/1755-8794-4-31.

Dolfsma, W. (2011). Patent strategizing. *Journal of Intellectual Capital*, *12*(2), 168–178, https://doi.org/10.1108/14691931111123377.

Domhan, T., Springenberg, T., & Hutter, F. (2015). Speeding Up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves. In *Proceedings - 24th international joint conference on artificial intelligence (IJCAI 2015)* https://www.aaai.org/ocs/index.php/IJCAI/IJCAI15/paper/viewPaper/11468.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78–87, https://doi.org/10.1145/2347736.2347755 https://dl.acm.org/doi/10.1145/2347736.2347755.

Du, S. (2019). Demystifying Focal Loss I: A More Focused Cross Entropy Loss https://medium.com/ai-salon/demystifying-focal-loss-i-a-more-focused-version-of-cross-entropy-loss-f49e4b044213.

Duguet, E. & MacGarvie, M. (2005). How well do patent citations measure flows of technology? Evidence from French innovation surveys. *Economics of Innovation and New Technology*, *14*(March 2015), 375–393, https://doi.org/10.1080/1043859042000307347.

Eclipse Deeplearning4j Development Team (2020). Deeplearning4j: Open-source distributed deep learning for the JVM http://deeplearning4j.org.

EPO (2016). *India and Europe explore the impact of Industry 4.0 on the patent system*. Technical report, European Patent Office, Munich, Germany.

EPO (2017). Cooperative Patent Classification (CPC).

Ernst, H. (1995). Patenting strategies in the German mechanical engineering industry and their relationship to company performance. *Technovation*, *15*(4), 225–240, https://doi.org/10.1016/0166-4972(95)96605-S.

Ernst, H. (2001). Patent applications and subsequent changes of performance: Evidence from time-series cross-section analyses on the firm level. *Research Policy*, *30*(1), 143–157, https://doi.org/10.1016/S0048-7333(99)00098-0.

Ernst, H. (2003). Patent information for strategic technology management. *World Patent Information*, *25*(3), 233–242, https://doi.org/10.1016/S0172-2190(03)00077-2.

Ernst, H., Legler, S., & Lichtenthaler, U. (2010). Determinants of patent value: Insights from a simulation analysis. *Technological Forecasting and Social Change*, *77*(1), 1–19, https://doi.org/10.1016/j.techfore.2009.06.009.

Ernst, H. & Omland, N. (2011). The Patent Asset Index - A new approach to benchmark patent portfolios. *World Patent Information*, *33*(1), 34–41, https://doi.org/10.1016/j.wpi.2010.08.008.

European Commission (2004). Innovation Management and the Knowledge-Driven Economy. *European Commission Directorate-general for Enterprise*, *4*(1), 164, https://doi.org/10.1016/j.nepr.2004.01.002.

Fagerberg, J., Mowery, D., & Nelson, R. (2006). *The Oxford Handbook of Innovation*. Oxford: Oxford University Press, 1st edition.

Falagas, M. E., Pitsouni, E. I., Malietzis, G. A., & Pappas, G. (2008). Comparison of PubMed, Scopus, Web of Science, and Google Scholar: Strengths and weaknesses. *FASEB Journal*, *22*(2), 338–342, https://doi.org/10.1096/fj.07-9492LSF.

Falk, N. & Train, K. (2017). Patent Valuation with Forecasts of Forward Citations. *Journal of Business Valuation and Economic Loss Analysis*, *12*(1), 101–121, https://doi.org/10.1515/jbvela-2016-0002.

Ferri, C., Hernández-Orallo, J., & Flach, P. (2019). Setting decision thresholds when operating conditions are uncertain. *Data Mining and Knowledge Discovery*, *33*(4), 805–847, https://doi.org/10.1007/s10618-019-00613-7.

Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, *30*(1), 27–38, https://doi.org/10.1016/j.patrec.2008.08.010.

Fischer, T. & Leidinger, J. (2014). Testing patent value indicators on directly observed patent value - An empirical analysis of Ocean Tomo patent auctions. *Research Policy*, *43*(3), 519–529, https://doi.org/10.1016/j.respol.2013.07.013.

Fonseca, E. (2019). State-of-the-art Multilingual Lemmatization https://towardsdatascience.com/state-of-the-art-multilingual-lemmatization-f303e8ff1a8.

Frietsch, R., Neuhäusler, P., Jung, T., & Van Looy, B. (2014). Patent indicators for macroeconomic growth - The value of patents estimated by export volume. *Technovation*, *34*(9), 546–558, https://doi.org/10.1016/j.technovation.2014.05.007.

Frietsch, R., et al. (2010). *The Value and Indicator Function of Patents*. Technical report, Fraunhofer Institute for Systems and Innovation Research http://www.e-fi.de/fileadmin/Studien/Studien_2010/15_2010_Patent_Value.pdf.

Gal, Y. & Ghahramani, Z. (2015). Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference. In *ICLR 2016* (pp. 1–12). http://arxiv.org/abs/1506.02158.

Gal, Y. & Ghahramani, Z. (2016). A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. In *30th Conference on Neural Information Processing Systems (NIPS 2016)* Barcelona, Spain.

Gambardella, A. (2011). *Innovative Science and Technology indicators combining patent data and surveys: Empirical models and policy analyses*. Technical Report SP1-Cooperation-SSH-CT-2008, 7th Framework Programme, Boconni University, Italy.

Gambardella, A., Giuri, P., & Luzzi, A. (2007). The market for patents in Europe. *Research Policy*, *36*(8), 1163–1183, https://doi.org/10.1016/j.respol.2007.07.006.

Gambardella, A., et al. (2005). *The Value of European Patents: Evidence from a survey of European Inventors*. Technical report, PatVal EU project, European Commission.

Gambardella, A., Harhoff, D., & Verspagen, B. (2011). The determinants of the private value of patents. *WIPO Economics and Statistics Series*.

Gay, C., Latham, W., & Le Bas, C. (2008). Collective knowledge, prolific inventors and the value of inventions: An empirical study of French, German and British patents in the US, 1975-1999. *Economics of Innovation and New Technology*, *17*(1), 5–22, https://doi.org/10.1080/10438590701279193.

Github (2020). Github https://github.com/.

Giuri, P., et al. (2007). Inventors and invention processes in Europe: Results from the PatVal-EU survey. *Research Policy*, *36*(8), 1107–1127, https://doi.org/10.1016/j.respol.2007.07.008.

Golstein, B. (2018). A Brief Taxonomy of AI https://www.sharper.ai/taxonomy-ai/.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press (Adaptive computation and machine learning series).

Google (2019). Google Patent Landscaping https://github.com/google/patents-public-data/tree/master/models/landscaping.

Google (2020a). Google AI Platform https://cloud.google.com/ai-platform.

Google (2020b). Patent analysis using the Google Patents Public Datasets on BigQuery https://github.com/google/patents-public-data.

Google Cloud (2019a). Expanding your patent set with ML and BigQuery https://cloud.google.com/blog/products/data-analytics/expanding-your-patent-set-with-ml-and-bigquery.

Google Cloud (2019b). Google Patent Landscaping Bucket https://console.cloud.google.com/storage/browser/patent_landscapes/.

Google Code (2013). word2vec https://word2vec.googlecode.com/svn/trunk/questions-words.txt%0A.

Govindarajan, U. H., Trappey, A. J., & Trappey, C. V. (2018). Immersive Technology for Human-Centric Cyberphysical Systems in Complex Manufacturing Processes: A Comprehensive Overview of the Global Patent Profile Using Collective Intelligence. *Hindawi Complexity*, *2018*, 1–18, https://doi.org/10.1155/2018/4283634.

Govindarajan, U. H., Trappey, A. J., & Trappey, C. V. (2019a). Intelligent collaborative patent mining using excessive topic generation. *Advanced Engineering Informatics*, *42*(June), 100955, https://doi.org/10.1016/j.aei.2019.100955.

Govindarajan, U. H., Trappey, A. J. C., & Trappey, C. V. (2019b). 360 ° Technology as a Gateway for Immersive Psychotherapy Applications : An Intelligent Patent Mining Analysis. In *International Conference on Data Science* (pp. 215–218).

Gramacy, R. B. & Taddy, M. (2010). Categorical inputs, sensitivity analysis, optimization and importance tempering with tgp version 2, an R package for treed gaussian process models. *Journal of Statistical Software*, *33*(6), 1–48, https://doi.org/10.18637/jss.v033.i06.

Granstrand, O. (1999). *The Economics and Management of Intellectual Property: Towards Intellectual Capitalism*. E. Elgar, 1999, illustrate edition https://www.e-elgar.com/shop/gbp/the-economics-and-management-of-intellectual-property-9781840644630.html.

Grant, R. M. (1991). The Resource-Based Theory of Competitive Advantage: Implications for Strategy Formulation. *California Management Review*, *33*(3), 114–135, https://doi.org/10.2307/41166664 http://journals.sagepub.com/doi/10.2307/41166664.

Grant, R. M. (2012). *Contemporary Strategy Analysis: Text and Cases*, volume 8. Wiley http://www.amazon.co.uk/Contemporary-Strategy-Analysis-Text-Cases/dp/0470747099.

Greenhalgh, C. & Rogers, M. (2006). The value of innovation: The interaction of competition, R&D and IP. *Research Policy*, *35*(4), 562–580, https://doi.org/10.1016/j.respol.2006.02.002.

Gregory, M. J. (1995). Technology Management: A Process Approach. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *209*(5), 347–356, https://doi.org/10.1243/PIME_PROC_1995_209_094_02.

Griliches, Z. (1998). *Patent statistics as economic indicators: a survey*, volume I.

Griliches, Z., Hall, B. H., & Pakes, A. (1991). R&D, patents, and market value revisited: Is there a second (technological opportunity) factor? *Economics of Innovation and New Technology*, *1*(3), 183–201, https://doi.org/10.1080/10438599100000001.

Grimaldi, M. & Cricelli, L. (2019). Indexes of patent value: a systematic literature review and classification. *Knowledge Management Research and Practice*, (pp. 1–20)., https://doi.org/10.1080/14778238.2019.1638737.

Grimaldi, M., Cricelli, L., Di Giovanni, M., & Rogo, F. (2015). The patent portfolio value analysis: A new framework to leverage patent information for strategic technology planning. *Technological Forecasting and Social Change*, *94*(1), 286–302, https://doi.org/10.1016/j.techfore.2014.10.013.

Grimaldi, M., Cricelli, L., & Rogo, F. (2018). Valuating and analyzing the patent portfolio: the patent portfolio value index. *European Journal of Innovation Management*, *21*(2), 174–205, https://doi.org/10.1108/EJIM-02-2017-0009.

Groeneveld, R. A. & Meeden, G. (1984). Measuring Skewness and Kurtosis. *Journal of the Royal Statistical Society. Series D (The Statistician)*, *33*(4), 391–399 http://www.jstor.org/stable/2987742.

Grupp, H. & Mogee, M. E. (2004). Indicators for national science and technology policy: How robust are composite indicators? *Research Policy*, *33*(9), 1373–1384, https://doi.org/10.1016/j.respol.2004.09.007.

Grzegorczyk, K. (2019). Vector representations of text data in deep learning http://arxiv.org/abs/1901.01695.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660, https://doi.org/10.1016/j.future.2013.01.010.

Guderian, C. C. (2019). Identifying Emerging Technologies with Smart Patent Indicators: The Example of Smart Houses. *International Journal of Innovation and Technology Management*, *16*(2), 1–24, https://doi.org/10.1142/S0219877019500408.

Günther, W. A., Rezazade Mehrizi, M. H., Huysman, M., & Feldberg, F. (2017). Debating big data: A literature review on realizing value from big data. *Journal of Strategic Information Systems*, *26*(3), 191–209, https://doi.org/10.1016/j.jsis.2017.07.003.

Gupta, K. (2000). Neural Network Structures. In *Neural Networks for RF and Microwave Design* (pp. 61–103).

Habibi, M., Weber, L., Neves, M., Wiegandt, D. L., & Leser, U. (2017). Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, *33*(14), i37–i48, https://doi.org/10.1093/bioinformatics/btx228.

Hagan, M. T., Demuth, H. B., Beale, M. H., & De Jesus, O. (1995). *Neural Network Design*, vol.2. Texas, USA: Boston Massachusetts PWS, 2nd edition.

Hagedoorn, J. & Cloodt, M. (2003). Measuring innovative performance: Is there an advantage in using multiple indicators? *Research Policy*, *32*(8), 1365–1379, https://doi.org/10.1016/S0048-7333(02)00137-3.

Haibo He & Garcia, E. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284, https://doi.org/10.1109/TKDE.2008.239 http://ieeexplore.ieee.org/document/5128907/.

Hall, B., Jaffe, A., & Trajtenberg, M. (2001). The NBER Patent Citations Data File: Lessons, Insights and Methodological Tools. (pp. 1–74)., https://doi.org/10.1186/1471-2164-12-148.

Hall, B. H. (2005). A Note on the Bias in Herfindahl-type Measures Based on Count Data. *2000*(September 2000), 1–10.

Hall, B. H., Jaffe, A., & Trajtenberg, M. (2005). Market Value and Patent Citations. *RAND Journal of Economics*, *36*(1), 16–38.

Hall, B. H. & MacGarvie, M. (2010). The private value of software patents. *Research Policy*, *39*(7), 994–1009, https://doi.org/10.1016/j.respol.2010.04.007.

Hall, B. H., Mairesse, J., & Mohnen, P. (2010). Chapter 24 - Measuring the Returns to RD. In B. H. Hall & N. Rosenberg (Eds.), *Handbook of the Economics of Innovation, Volume 2*, volume 2 of *Handbook of the Economics of Innovation* (pp. 1033 – 1082). North-Holland http://www.sciencedirect.com/science/article/pii/S0169721810020083.

Han, Q., Heimerl, F., Codina-Filba, J., Lohmann, S., Wanner, L., & Ertl, T. (2017). Visual patent trend analysis for informed decision making in technology management. *World Patent Information*, *49*, 34–42, https://doi.org/10.1016/j.wpi.2017.04.003.

Har-Peled, S., Roth, D., & Zimak, D. (2002). Constraint Classification: a new approach for Multiclass Classification and Ranking. In *NIPS 2002* http://papers.nips.cc/paper/2295-constraint-classification-for-multiclass-classification-and-ranking.pdf.

Harabi, N. (1995). Appropriability of technical innovations an empirical analysis. *Research Policy*, *24*(6), 981–992, https://doi.org/10.1016/0048-7333(94)00812-4.

Harhoff, D., Hall, B. H., Graevenitz von, G., Hoisl, K., Wagner, S., Gambardella, A., & Giuri, P. (2007). *The strategic use of patents and its implications for enterprise and competition policies*. Technical report, European Commission Ref. Ares(2014)78204.

Harhoff, D. & Hoisl, K. (2006). Everything you Always Wanted to Know About Inventors (But Never Asked): Evidence from the PatVal-EU Survey Discussion. *Discussion Papers in Business Administration, No. 2006-11*, (pp. 1–46). http://hdl.handle.net/10419/104473.

Harhoff, D., Narin, F., Scherer, F. M., & Vopel, K. (1999). Citation Frequency and the Value of Patented Inventions. *Review of Economics and Statistics*, *81*(August), 511–515, https://doi.org/10.1162/003465399558265.

Harhoff, D. & Reitzig, M. (2004). Determinants of opposition against EPO patent grants - The case of biotechnology and pharmaceuticals. *International Journal of Industrial Organization*, *22*(4), 443–480, https://doi.org/10.1016/j.ijindorg.2004.01.001.

Harhoff, D., Scherer, F. M., & Vopel, K. (2003). Citations, family size, opposition and the value of patent rights. *Research Policy*, *32*, 1343–1363, https://doi.org/10.1016/S0048-7333(02)00124-5.

Harhoff, D. & Wagner, S. (2009). The duration of patent examination at the european patent office. *Management Science*, *55*(12), 1969–1984, https://doi.org/10.1287/mnsc.1090.1069.

Harrel, F. (2019). Classification vs. Prediction https://www.fharrell.com/post/classification/.

Hartman, P., Bezos, J. P., Kaphan, S., & Spiegel, J. (1999). Method and system for placing a purchase order via a communications network http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.html&r=1&f=G&l=50&d=PALL&s1=5960411.PN.

Hastie, Trevor, Tibshirani, Robert, Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics.

Heaton, J. (2017). The Number of Hidden Layers https://www.heatonresearch.com/2017/06/01/hidden-layers.html.

Heaton, J. T. (2005). *Introduction to Neural Networks with Java*. Heaton Research.

Helmers, L., et al. (2019). Automating the search for a patent's prior art with a full text similarity search. *PLoS ONE*, *14*(3), 1–23, https://doi.org/10.1371/journal.pone.0212103.

Hess, P. K. (2008). Enforcing and Challenging Intellectual Property Rights. *ip4inno.eu*, (March), 1–27.

Hido, S., et al. (2012). Modeling Patent Quality: A System for Large-scale Patentability Analysis using Text Mining. *Journal of Information Processing*, *20*(3), 655–666, https://doi.org/10.2197/ipsjjip.20.655.

Hill, T., Marquez, L., O'Connor, M., & Remus, W. (1993). Artificial neural network models for forecasting and decision making. *International Journal of Forecasting*, *1*(1), 5–15, https://doi.org/10.1016/0169-2070(94)90045-0.

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554, https://doi.org/10.1162/neco.2006.18.7.1527.

Hoerl, A. E. & Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, https://doi.org/10.1080/00401706.1970.10488634.

Hofstätter, S., Rekabsaz, N., Lupu, M., Eickhoff, C., & Hanbury, A. (2019). Enriching word embeddings for patent retrieval with global context. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11437 LNCS*, 810–818, https://doi.org/10.1007/978-3-030-15712-8_57.

Honkela, A. (2001). Multilayer perceptrons https://users.ics.aalto.fi/ahonkela/dippa/node41.html.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, *79*(8), 2554–2558, https://doi.org/10.1073/pnas.79.8.2554.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*(2), 251–257, https://doi.org/10.1016/0893-6080(91)90009-T.

Hsieh, C. H. (2013). Patent value assessment and commercialization strategy. *Technological Forecasting and Social Change*, *80*(2), 307–319, https://doi.org/10.1016/j.techfore.2012.09.014.

Hu, J., Li, S., Hu, J., & Yang, G. (2018a). A hierarchical feature extraction model for multi-label mechanical patent classification. *Sustainability (Switzerland)*, *10*(1), https://doi.org/10.3390/su10010219.

Hu, J., Li, S., Yao, Y., Yu, L., Yang, G., & Hu, J. (2018b). Patent keyword extraction algorithm based on distributed representation for patent classification. *Entropy*, *20*(2), https://doi.org/10.3390/e20020104.

Hudson, P. T. W. & Postma, E. O. (1982). Choosing and using neural net. *Working paper article*, (pp. 1–15).

Hurtado, J. L., Agarwal, A., & Zhu, X. (2016). Topic discovery and future trend forecasting for texts. *Journal of Big Data*, *3*(1), https://doi.org/10.1186/s40537-016-0039-2.

Ilevbare, I., Dusch, B., & Templeton, P. (2016). *A framework and methodology for creating business tools*. Technical report.

Ioffe, S. & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *32nd International Conference on Machine Learning, ICML 2015*, volume 1 (pp. 448–456).

Ippolito, P. P. (2019). Hyperparameters Optimization https://towardsdatascience.com/hyperparameters-optimization-526348bb8e2d.

Jääsaari, E., Hyvönen, V., & Roos, T. (2019). Efficient autotuning of hyperparameters in approximate nearest neighbor search. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11440 LNAI*, 590–602, https://doi.org/10.1007/978-3-030-16145-3_46.

Jain, R. (2019). Activation Functions in Neural Networks https://www.linkedin.com/pulse/activation-functions-neural-networks-rahul-jain.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*, volume 103 of *Springer Texts in Statistics*. New York, NY: Springer New York http://link.springer.com/10.1007/978-1-4614-7138-7.

Jane Street (2019). L2 Regularization and Batch Norm https://blog.janestreet.com/l2-regularization-and-batch-norm/.

Jensen, P. H., Thomson, R., & Yong, J. (2011). Estimating the patent premium: Evidence from the Australian Inventor Survey. *Strategic Management Journal*, *32*(10), 1128–1138, https://doi.org/10.1002/smj.925.

Jeong, Y., Aristodemou, L., & Tietze, F. (2019). Exploring disruptive innovation opportunity using patent analysis and deep learning. In *R&D Management Conference 2019, Paris, France* (pp. 1–14).

Jeong, Y., Park, I., & Yoon, B. (2018). Identifying emerging Research and Business Development (R&BD) areas based on topic modeling and visualization with intellectual property right data. *Technological Forecasting and Social Change*, *146*(January 2018), 655–672, https://doi.org/10.1016/j.techfore.2018.05.010.

Jia, F., Lei, Y., Lin, J., Zhou, X., & Lu, N. (2016). Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mechanical Systems and Signal Processing*, *72-73*, 303–315, https://doi.org/10.1016/j.ymssp.2015.10.025.

Jokanović, B., Lalic, B., Milovančević, M., Simeunović, N., & Marković, D. (2017). Economic development evaluation based on science and patents. *Physica A: Statistical Mechanics and its Applications*, *481*, 141–145, https://doi.org/10.1016/j.physa.2017.04.015.

Jun, S. (2013). Examining technological innovation of Apple using patent analysis. *Industrial Management & Data Systems*, *113*, 890–907, https://doi.org/10.1108/IMDS-01-2013-0032.

Jun, S. & Lee, S.-j. (2012). Emerging Technology Forecasting Using New Patent Information Analysis. *International Journal of Software Engineering and Its Applications*, *6*(3), 107–116.

Jun, S., Park, S. S., & Jang, D. S. (2014). Document clustering method using dimension reduction and support vector clustering to overcome sparseness. *Expert Systems with Applications*, *41*(7), 3204–3212, https://doi.org/10.1016/j.eswa.2013.11.018.

Jurafsky, D. & Martin, J. H. (2016). Semantics with Dense Vectors. *Speech and Language Processing, 3rd edition*.

Kalutkiewicz, M. J. & Ehman, R. L. (2014). Patents as proxies: NIH hubs of innovation. *Nature Biotechnology*, *32*(6), 536–537, https://doi.org/10.1038/nbt.2917.

Kapil, D. (2018). Focal Loss Demystified https://medium.com/adventures-with-deep-learning/focal-loss-demystified-c529277052de.

Kapoor, R., Karvonen, M., & Kässi, T. (2013). Patent value indicators as proxy for commercial value of inventions. *International Journal of Intellectual Property Management*, *6*(3), 217–232, https://doi.org/10.1504/IJIPM.2013.056242.

Karanikić, P., Mladenović, I., Sokolov-Mladenović, S., & Alizamir, M. (2017). Prediction of economic growth by extreme learning approach based on science and technology transfer. *Quality & Quantity*, *51*(3), 1395–1401, https://doi.org/10.1007/s11135-016-0337-y http://link.springer.com/10.1007/s11135-016-0337-y.

Karpathy, A. (2017). A Peek at Trends in Machine Learning https://medium.com/@karpathy/a-peek-at-trends-in-machine-learning-ab8a1085a106.

Karpathy, A. & Lei, F.-F. (2015). CS231n Convolutional Neural Networks for Visual Recognition http://vision.stanford.edu/teaching/cs231n/.

Kathuria, A. (2018). How to chose an activation function for your network https://blog.paperspace.com/vanishing-gradients-activation-function/.

Kavzoglu, T. & Mather, P. M. (2003). The use of backpropagating artificial neural networks in land cover classification. *International Journal of Remote Sensing*, *24*(23), 4907–4938, https://doi.org/10.1080/0143116031000114851.

Khachatryan, D. & Muehlmann, B. (2019). Measuring technological breadth and depth of patent documents using Rao's Quadratic Entropy. *Journal of Applied Statistics*, *0*(0), 1–26, https://doi.org/10.1080/02664763.2019.1619072.

Kim, J. & Lee, S. (2017). Forecasting and identifying multi-technology convergence based on patent data: the case of IT and BT industries in 2020. *Scientometrics*, *111*(1), 47–65, https://doi.org/10.1007/s11192-017-2275-4.

Kim, J., Yoon, J., Park, E., & Choi, S. (2020). Patent document clustering with deep embeddings. *Scientometrics*, *123*(2), 563–577, https://doi.org/10.1007/s11192-020-03396-7.

King, G. & Zeng, L. (2003). Logistic regression in rare events data. *Journal of Statistical Software*, *8*, 137–163, https://doi.org/10.1093/oxfordjournals.pan.a004868.

Kingma, D. P. & Ba, J. L. (2015). ADAM: A method for stochastic optimization. In *ICLR 2015* (pp. 1–15).

Klinger, R., Kolářik, C., Fluck, J., Hofmann-Apitius, M., & Friedrich, C. M. (2008). Detection of IUPAC and IUPAC-like chemical names. *Bioinformatics*, *24*(13), 268–276, https://doi.org/10.1093/bioinformatics/btn181.

Knet.jl (2020). Softmax Classification https://knet.readthedocs.io/en/latest/softmax.html.

Koehrsen, W. (2018a). Automated Machine Learning Hyperparameter Tuning in Python https://towardsdatascience.com/automated-machine-learning-hyperparameter-tuning-in-python-dfda59b72f8a.

Koehrsen, W. (2018b). Beyond Accuracy: Precision and Recall https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c.

Koehrsen, W. (2018c). Feature Engineering: What Powers Machine Learning https://towardsdatascience.com/feature-engineering-what-powers-machine-learning-93ab191bcc2d.

Kogan, L., Papanikolaou, D., Seru, A., & Stoffman, N. (2017). Technological Innovation, Resource Allocation and Growth. *The Quarterly Journal of Economics*, *132*(2), 665–712, https://doi.org/10.1093/qje/qjw040.

Kohavi, R. (1995). *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection*. Technical report.

Komer, B., Bergstra, J., & Eliasmith, C. (2014). *Hyperopt-Sklearn: Automatic Hyperparameter Configuration for Scikit-Learn*. Technical report.

Krallinger, M., Leitner, F., Rabal, O., Vazquez, M., Oyarzabal, J., & Valencia, A. (2015). CHEMDNER: The drugs and chemical names extraction challenge. *Journal of Cheminformatics*, *7*(Suppl 1), 1–11, https://doi.org/10.1186/1758-2946-7-S1-S1.

Kuhn, M. & Johnson, K. (2013). *Applied predictive modeling*.

Kukačka, J., Golkov, V., & Cremers, D. (2017). Regularization for Deep Learning: A Taxonomy. (pp. 1–23). http://arxiv.org/abs/1710.10686.

Kumar, S. (2020). Data splitting technique to fit any Machine Learning Model https://towardsdatascience.com/data-splitting-technique-to-fit-any-machine-learning-model-c0d7f3f1c790.

Kwag, C. (2018). Focal Loss (a.k.a RetinaNet) paper review https://chadrick-kwag.net/focal-loss-a-k-a-retinanet-paper-review/.

Kyebambe, M. N., Cheng, G., Huang, Y., He, C., & Zhang, Z. (2017). Forecasting emerging technologies: A supervised learning approach through patent analysis. *Technological Forecasting and Social Change*, *125*(July), 236–244, https://doi.org/10.1016/j.techfore.2017.08.002.

Labach, A., Salehinejad, H., & Valaee, S. (2019). Survey of Dropout Methods for Deep Neural Networks http://arxiv.org/abs/1904.13310.

Lagrost, C., Martin, D., Dubois, C., & Quazzotti, S. (2010). Intellectual property valuation: How to approach the selection of an appropriate valuation method. *Journal of Intellectual Capital*, *11*(4), 481–503, https://doi.org/10.1108/14691931011085641.

Lai, Y. H. & Che, H. C. (2009a). Evaluating patents using damage awards of infringement lawsuits: A case study. *Journal of Engineering and Technology Management*, *26*(3), 167–180, https://doi.org/10.1016/j.jengtecman.2009.06.005.

Lai, Y. H. & Che, H. C. (2009b). Modeling patent legal value by Extension Neural Network. *Expert Systems with Applications*, *36*(7), 10520–10528, https://doi.org/10.1016/j.eswa.2009.01.027 http://dx.doi.org/10.1016/j.eswa.2009.01.027.

Lamirel, J.-C., Al Shehabi, S., Hoffmann, M., & François, C. (2003). Intelligent patent analysis through the use of a neural network http://dl.acm.org/citation.cfm?id=1119303.1119305.

Lanjouw, J. & Schankerman, M. (2001). Characteristics of Patent Litigation: A Window on Competition. *RAND Journal of Economics*, *32*(1), 129–151.

Lanjouw, J. O., Pakes, A., & Putnam, J. (1998). How to count patents and value intellectual property: The uses of patent renewal and application data. *Journal of Industrial Economics*, *46*(4), 405–432, https://doi.org/10.1111/1467-6451.00081.

Lanjouw, J. O. & Schankerman, M. (2004). Patent quality and research productivity: Measuring innovation with multiple indicators. *Economic Journal*, *114*(495), 441–465, https://doi.org/10.1111/j.1468-0297.2004.00216.x.

Lau, S. (2017). Learning Rate Schedules and Adaptive Learning Rate Methods for Deep Learning https://towardsdatascience.com/learning-rate-schedules-and-adaptive-learning-rate-methods-for-deep-learning-2c8f433990d1.

Lau, S., Gonzalez, J., & Nolan, D. (2020). *Principles and Techniques of Data Science*, volume DS100 Data. UC Berkeley https://www.textbook.ds100.org/intro.html.

Le, Q., Mikolov, T., & Com, T. G. (2014). Distributed Representations of Sentences and Documents. *Google Research working paper series*.

Leaman, R., Wei, C. H., Zou, C., & Lu, Z. (2016). Mining chemical patents with an ensemble of open systems. *Database : the journal of biological databases and curation*, *2016*(December), 1–7, https://doi.org/10.1093/database/baw065.

Leamer, E. E. (1985). Sensitivity Analyses Would Help. *The American Economic Review*, *75*(3), 308–313 http://www.jstor.org/stable/1814801.

LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K. R. (2012). Efficient backprop. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, https://doi.org/10.1007/978-3-642-35289-8-3.

Lee, C., Jeon, J., & Park, Y. (2011). Monitoring trends of technological changes based on the dynamic patent lattice: A modified formal concept analysis approach. *Technological Forecasting and Social Change*, *78*(4), 690–702, https://doi.org/10.1016/j.techfore.2010.11.010.

Lee, C., Kim, J., Kwon, O., & Woo, H. G. (2016). Stochastic technology life cycle analysis using multiple patent indicators. *Technological Forecasting and Social Change*, *106*, 53–64, https://doi.org/10.1016/j.techfore.2016.01.024.

Lee, C., Kwon, O., Kim, M., & Kwon, D. (2018). Early identification of emerging technologies: A machine learning approach using multiple patent indicators. *Technological Forecasting and Social Change*, *127*(April 2017), 291–303, https://doi.org/10.1016/j.techfore.2017.10.002.

Lee, C., Song, B., & Park, Y. (2013). How to assess patent infringement risks: a semantic patent claim analysis using dependency relationships. *Technology Analysis & Strategic Management*, *25*(1), 23–38, https://doi.org/10.1080/09537325.2012.748893.

Lee, J., Jang, D., & Park, S. (2017). Deep learning-based corporate performance prediction model considering technical capability. *Sustainability (Switzerland)*, *9*(6), 1–12, https://doi.org/10.3390/su9060899.

Lee, J., Xiao, L., Schoenholz, S. S., Bahri, Y., Novak, R., Sohl-Dickstein, J., & Pennington, J. (2019). Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent http://arxiv.org/abs/1902.06720.

Lee, J.-S. & Hsiang, J. (2019a). Patent Claim Generation by Fine-Tuning BERT Language Model. *Working paper article*, *1*, 1–6 http://arxiv.org/abs/1906.02124.

Lee, J.-S. & Hsiang, J. (2019b). Patent Claim Generation by Fine-Tuning OpenAI GPT-2. *Working paper article*, (pp. 1–11). http://arxiv.org/abs/1907.02052.

Lee, Y. C., Lapedes, A., & Farber, R. (1989). How Neural Nets Work. In *Evolution, Learning and Cognition* (pp. 331–346).

Lei, L., Qi, J., & Zheng, K. (2019). Patent analytics based on feature vector space model: A case of IoT. *IEEE Access*, *7*, 45705–45715, https://doi.org/10.1109/ACCESS.2019.2909123.

Leonard, N. (2013). word2vec questions-words. *Google Research* https://github.com/nicholas-leonard/word2vec/blob/master/questions-words.txt.

Leten, B., Belderbos, R., & Van Looy, B. (2007). Technological diversification, coherence, and performance of firms. *Journal of Product Innovation Management*, *24*(6), 567–579, https://doi.org/10.1111/j.1540-5885.2007.00272.x.

Li, S., Hu, J. J., Cui, Y., & Hu, J. J. (2018). DeepPatent: patent classification with convolution neural networks and word embedding. *Scientometrics*, *117*(2), 721–744, https://doi.org/10.1007/s11192-018-2905-5.

Li, X., Aristodemou, L., Tietze, F., & Jeong, Y. (2020). Disruptive technologies: characteristics and early identification, using machine learning and text mining from patent data. In *R&D Management Conference 2020, University of Strathclyde, Glasgow, Scotland, United Kingdom* (pp. Conference postponed due to COVID–19).

Li, X., Chen, H., Zhang, Z., Li, J., & Nunamaker, J. F. (2009). Managing Knowledge in Light of Its Evolution Process: An Empirical Study on Citation Network-Based Patent Classification. *Journal of Management Information Systems*, *26*(1), 129–154, https://doi.org/10.2753/MIS0742-1222260106.

Lin, B. W., Chen, C. J., & Wu, H. L. (2006). Patent portfolio diversity, technology strategy, and firm value. *IEEE Transactions on Engineering Management*, *53*(1), 17–26, https://doi.org/10.1109/TEM.2005.861813.

Lin, B. W. & Chen, J. S. (2005). Corporate technology portfolios and R&D performance measures: A study of technology intensive firms. *R and D Management*, *35*(2), 157–170, https://doi.org/10.1111/j.1467-9310.2005.00380.x.

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2017). Focal Loss for Dense Object Detection. *Proceedings of the IEEE International Conference on Computer Vision*, *2017-Octob*, 2999–3007, https://doi.org/10.1109/ICCV.2017.324.

Linderman, G. C. & Steinerberger, S. (2019). Clustering with t-SNE, Provably. *SIAM Journal on Mathematics of Data Science*, *1*(2), 313–332, https://doi.org/10.1137/18m1216134.

Lipton, Z. C., Elkan, C., Naryanaswamy, B., & Narayanaswamy, B. (2014). Thresholding Classifiers to Maximize F1 Score http://arxiv.org/abs/1402.1892https://www.groundai.com/project/thresholding-classifiers-to-maximize-f1-score/2.

Liu, H. & Motoda, H. (1999). Feature Extraction Construction and Selection: A Data Mining Perspective. *Journal of the American Statistical Association*, https://doi.org/10.2307/2669967.

Liu, K., Arthurs, J., Cullen, J., & Alexander, R. (2008). Internal sequential innovations: How does interrelatedness affect patent renewal? *Research Policy*, *37*(5), 946–953, https://doi.org/10.1016/j.respol.2008.03.005.

Lj Miranda (2017). Understanding softmax and the negative log-likelihood. *Lj Miranda Blog* https://ljvmiranda921.github.io/notebook/2017/08/13/softmax-and-the-negative-log-likelihood/.

Lorena, A. C., De Carvalho, A. C., & Gama, J. M. (2008). A review on the combination of binary classifiers in multiclass problems. *Artificial Intelligence Review*, *30*(1-4), 19–37, https://doi.org/10.1007/s10462-009-9114-9.

Loshchilov, I. & Hutter, F. (2019a). Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019*, volume 100.

Loshchilov, I. & Hutter, F. (2019b). SGDR: Stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings* (pp. 1–16).

Lu, B., Wang, X., & Utiyama, M. (2009). Incorporating prior knowledge into learning by dividing training data. *Frontiers of Computer Science in China*, *3*(1), 109–122, https://doi.org/10.1007/s11704-009-0013-7.

Lu, Y., Xiong, X., Zhang, W., Liu, J., & Zhao, R. (2020). Research on classification and similarity of patent citation based on deep learning. *Scientometrics*, (0123456789), https://doi.org/10.1007/s11192-020-03385-w.

Lupu, M. (2013). Patent Retrieval. *Foundations and Trends® in Information Retrieval*, *7*(1), 1–97, https://doi.org/10.1561/1500000027.

Lupu, M. (2018). Artificial Intelligence and Intellectual Property. *World Patent Information*, *53*, A1–A3, https://doi.org/10.1016/j.wpi.2018.06.001.

Lupu, M., Mayer, K., Tait, J., & Trippe, A. (2011). Current Challenges in Patent Information Retrieval. In M. Lupu, K. Mayer, J. Tait, & A. J. Trippe (Eds.), *The Information Retrieval Serives*, volume 29 of *The Information Retrieval Series* (pp. 415). Berlin, Heidelberg: Springer Berlin Heidelberg http://link.springer.com/10.1007/978-3-642-19231-9.

Ma, S.-C. S. C., Feng, L., Yin, Y., & Wang, J. (2019). Research on petroleum patent valuation based on Value Capture Theory. *World Patent Information*, *56*(18), 29–38, https://doi.org/10.1016/j.wpi.2018.10.004.

Maheswari, J. P. (2019). Breaking the curse of small data sets in Machine Learning: Part 2. *Towards Data Science* https://towardsdatascience.com/breaking-the-curse-of-small-data-sets-in-machine-learning-part-2-894aa45277f4.

Maklin, C. (2019). Dropout Neural Network Layer In Keras Explained. *Towards Data Science* https://towardsdatascience.com/machine-learning-part-20-dropout-keras-layers-explained-8c9f6dc4c9ab.

Manning, C. D., Raghavan, P., & Schutze, H. (2008). *Introduction to Information Retrieval*. Cambridge: Cambridge University Press http://ebooks.cambridge.org/ref/id/CBO9780511809071.

Mao, L. (2019). Use Focal Loss To Train Model Using Imbalanced Dataset. *Lei Mao's Log Book* https://leimao.github.io/blog/Focal-Loss-Explained/.

Marco, A. C., Sarnoff, J. D., & DeGrazia, C. A. W. (2019). Patent claims and patent scope. *Research Policy*, (March), 103790, https://doi.org/10.1016/j.respol.2019.04.014.

Maren, A. J. (1991). A Logical Topology of Neural Networks. *Second workshop on Neural Networks WNN-AIND91*, (pp.̃27). http://www.aliannajmaren.com/Downloads/Logical-topology-neural-networks.pdf.

Mariani, M. & Romanelli, M. (2007). "Stacking" and "picking" inventions: The patenting behavior of European inventors. *Research Policy*, *36*(8), 1128–1142, https://doi.org/10.1016/j.respol.2007.07.009.

Mariani, M. S., Medo, M., & Lafond, F. (2019). Early identification of important patents: Design and validation of citation network metrics. *Technological Forecasting and Social Change*, *146*(October 2017), 644–654, https://doi.org/10.1016/j.techfore.2018.01.036.

Marković, D. (2017). Appraisal of Science and Economic Factors on Total Number of Granted Patents. *Networks and Spatial Economics*, *18*(4), 1019–1026, https://doi.org/10.1007/s11067-017-9373-y.

Marković, D., Petković, D., Nikolić, V., Milovančević, M., & Petković, B. (2017). Soft computing prediction of economic growth based in science and technology factors. *Physica A: Statistical Mechanics and its Applications*, *465*, 217–220, https://doi.org/10.1016/j.physa.2016.08.034.

Martinez, C. (2010). Insight into Different Types of Patent Families Catalina Martinez. *OECD Science, Technology and Industry Working Papers*, *02*, https://doi.org/http://dx.doi.org/10.1787/5kml97dr6ptl-enOECD.

Martínez, C. (2011). Patent families: When do different definitions really matter? *Scientometrics*, *86*(1), 39–63, https://doi.org/10.1007/s11192-010-0251-3.

Mattyws Grawe, s. F., Claudia Martins, n. A., & Bonfante, A. G. (2017). Automated Patent Classification Using Word Embedding. https://doi.org/10.1109/ICMLA.2017.0-127.

McGonagle, J., Shaikouski, G., & Williams, G. (2020). Backpropagation. *Brilliant Math & Science Wiki* https://brilliant.org/wiki/backpropagation/.

McKinney, W. (2010). Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference* (pp. 56–61). https://conference.scipy.org/proceedings/scipy2010/mckinney.html.

Messeni Petruzzelli, A., Rotolo, D., & Albino, V. (2015). Determinants of patent citations in biotechnology: An analysis of patent influence across the industrial and organizational boundaries. *Technological Forecasting and Social Change*, *91*, 208–221, https://doi.org/10.1016/j.techfore.2014.02.018.

Mhaskar, H., Liao, Q., & Poggio, T. (2017). When and why are deep networks better than shallow ones? In *31st AAAI Conference on Artificial Intelligence, AAAI 2017* (pp. 2343–2349).

Mhaskar, H. N. & Poggio, T. (2016). Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, *14*(6), 829–848, https://doi.org/10.1142/S0219530516400042.

Microsoft (2020). Microsoft Azure https://azure.microsoft.com/en-gb/.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings* (pp. 1–12).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013b). Distributed Representations of Words and Phrases and their Compositionality. *Google Research working paper series*, (pp. 1–9)., https://doi.org/10.1162/jmlr.2003.3.4-5.951.

Milanez, D. H., Faria, L. I. L. d. L., do Amaral, R. M., & Gregolin, J. A. R. (2017). Claim-based patent indicators: A novel approach to analyze patent content and monitor technological advances. *World Patent Information*, *50*, 64–72, https://doi.org/10.1016/j.wpi.2017.08.008.

Miller, D. J. (2006). Technological diversity, related diversification, and firm performance. *Strategic Management Journal*, *27*(7), 601–619, https://doi.org/10.1002/smj.533.

ML Glossary (2020). Activation Functions. *ML Glossary documentation* https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html.

Moehrle, M. G., Walter, L., Bergmann, I., Bobe, S., & Skrzipale, S. (2010). Patinformatics as a business process: A guideline through patent research tasks and tools. *World Patent Information*, *32*(4), 291–299, https://doi.org/10.1016/j.wpi.2009.11.003.

Momeni, A. & Rost, K. (2016). Identification and monitoring of possible disruptive technologies by patent-development paths and topic modeling. *Technological Forecasting and Social Change*, *104*, 16–29, https://doi.org/10.1016/j.techfore.2015.12.003.

Mowery, D. C., Oxley, J. E., & Silverman, B. S. (1996). Strategic alliances and interfirm knowledge transfer. *Strategic Management Journal*, *17*(S2), 77–91, https://doi.org/10.1002/smj.4250171108.

Mowery, D. C. & Rosenberg, N. (1989). *Technology and the Pursuit of Economic Growth*. Cambridge University Press.

Mukhoti, J., Kulharia, V., Sanyal, A., Golodetz, S., Torr, P., & Dokania, P. (2020). The intriguing effects of focal loss on the calibration of deep neural networks. In *ICLR2020*, volume 38 (pp. 67–68).

Munari, F. & Oriani, R. (2011). *The economic valuation of patents: Methods and applications*. Edward Elgar.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT Press (Adaptive computation and machine learning series).

Narin, F. & Hamilton, K. S. (1996). Bibliometric performance measures. *Scientometrics*, *36*(3), 293–310, https://doi.org/10.1007/BF02129596.

Neuhäusler, P. & Frietsch, R. (2013). Patent families as macro level patent value indicators: Applying weights to account for market differences. *Scientometrics*, *96*(1), 27–49, https://doi.org/10.1007/s11192-012-0870-y.

Neyshabur, B., Bhojanapalli, S., McAllester, D., & Srebro, N. (2017). Exploring generalization in deep learning. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 5948–5957.

Ng, A. (2017). Backpropagation. *Stanford University CS229 Machine Learning*, (pp. 1–2). http://cs229.stanford.edu/notes/backprop.py.

Ng, A. & Katanforoosh, K. (2020). Splitting into train, dev and test sets http://cs230.stanford.edu/blog/split/.

Ng, A. Y. (2004). Feature selection, L1 vs. L2 regularisation, and rotational invariane. *Proceedings of the 21 st International Conference on Machine Learning*.

Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press.

Nieradzik, L. (2019). Losses for Image Segmentation. *Lars' Blog* https://lars76.github.io/neural-networks/object-detection/losses-for-segmentation/.

Noh, H. & Lee, S. (2020). Forecasting Forward Patent Citations: Comparison of Citation-Lag Distribution, Tobit Regression, and Deep Learning Approaches. *IEEE Transactions on Engineering Management*, (pp. 1–12).

Oakley, J. E. & O'Hagan, A. (2004). Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *66*(3), 751–769, https://doi.org/10.1111/j.1467-9868.2004.05304.x.

OECD (2008). *Handbook on Constructing Composite Indicators: Methodology and User Guide*. Technical report https://www.oecd-ilibrary.org/economics/handbook-on-constructing-composite-indicators-methodology-and-user-guide_9789264043466-en.

OECD (2009). *OECD Patent Statistics Manual* http://www.oecd-ilibrary.org/science-and-technology/oecd-patent-statistics-manual_9789264056442-en.

OECD (2017). *The Next Production Revolution: Implications for Governments and Business*. Technical report, OECD - The Next Production Revolution.

OECD (2019a). *Artificial Intelligence in Society*. Technical report, OECD Publishing - Going Digital: Shaping Policies, Improving Lives, Paris.

OECD (2019b). *Measuring the digital transformation: a roadmap for the future*. Technical report, OECD.

Oldham, G. R. & Fried, Y. (2016). Job design research and theory: Past, present and future. *Organizational Behavior and Human Decision Processes*, *136*, 20–35, https://doi.org/10.1016/j.obhdp.2016.05.002.

Oliinyk, H. (2017). Hierarchical softmax and negative sampling: short notes worth telling https://towardsdatascience.com/hierarchical-softmax-and-negative-sampling-short-notes-worth-telling-2672010dbe08.

Oppermann, A. (2020). Regularization in Deep Learning - L1, L2, and Dropout https://www.deeplearning-academy.com/p/ai-wiki-regularization.

Ozturk, T., Talo, M., Yildirim, E. A., Baloglu, U. B., Yildirim, O., & Rajendra Acharya, U. (2020). Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Computers in Biology and Medicine*, *121*(April), 103792, https://doi.org/10.1016/j.compbiomed.2020.103792.

Pagels, M. (2018). Machine Learning Reductions & Mother Algorithms, Part II: Multi-class to Binary Classification https://medium.com/value-stream-design/machine-learning-reductions-mother-algorithms-part-ii-multiclass-to-binary-classification-1dad599147b.

Pagliardini, M., Gupta, P., & Jaggi, M. (2018). : Unsupervised Learning of Sentence Embeddings Using Compositional n-Gram Features. (pp. 528–540). http://arxiv.org/abs/1703.02507.

Pal, S. K. & Mitra, S. (1992). Multilayer Perceptron, Fuzzy Sets, and Classification. *IEEE Transactions on Neural Networks*, *3*(5), 683–697, https://doi.org/10.1109/72.159058.

Pandey, G. & Dukkipati, A. (2014). To go deep or wide in learning? *Journal of Machine Learning Research*, *33*, 724–732.

Paré, G., Trudel, M. C., Jaana, M., & Kitsiou, S. (2015). Synthesizing information systems knowledge: A typology of literature reviews. *Information and Management*, *52*(2), 183–199, https://doi.org/10.1016/j.im.2014.08.008.

Parr, R. & Sullivan, P. (1996). *Technology Licensing: Corporate Strategies for Maximizing Value*. Wiley.

Pedregosa, F., et al. (2019). Visualizing cross-validation behaviour in scikit-learn — scikit-learn 0.23.1 documentation.

Pedregosa, F., et al. (2020). 3.3. Metrics and scoring: quantifying the quality of predictions — scikit-learn 0.23.2 documentation.

Pezzotti, N., et al. (2020). GPU Linear Complexity t-SNE Optimization. *IEEE Transactions on Visualization and Computer Graphics*, *26*(1), 1172–1181, https://doi.org/10.1109/TVCG.2019.2934307.

Phaal, R. (2004). Technology roadmapping - A planning framework for evolution and revolution. *Technological Forecasting and Social Change*, *71*(1-2), 5–26, https://doi.org/10.1016/S0040-1625(03)00072-6.

Phaal, R., Farrukh, C. J., & Probert, D. R. (2001). Technology Roadmapping: linking technology resources to business objectives. *International Journal of Technology Management*, *26*(1), 2, https://doi.org/10.1504/IJTM.2003.003140.

Phaal, R., Routley, M., Athanassopoulou, N., & Probert, D. (2012). Charting Exploitation Strategies for Emerging Technology. *Research-Technology Management*, *55*(2), 34–42, https://doi.org/10.5437/08956308X5502021.

Pitkethly, R. (1997). The valuation of patents : A review of patent valuation methods with consideration of option based methods and the potential for further research. *New Developments in Intellectual Property : Law and Economics*, *2*(March), 1–32, https://doi.org/10.5791/0882-2875-2.3.5.

Poege, F., Harhoff, D., Gaessler, F., & Baruffaldi, S. (2019). Science quality and the value of inventions. *Science Advances*, *5*(12), https://doi.org/10.1126/sciadv.aay7323.

Porter, A. L., Garner, J., Carley, S. F., & Newman, N. C. (2019). Emergence scoring to identify frontier R&D topics and key players. *Technological Forecasting and Social Change*, *146*(October 2017), 628–643, https://doi.org/10.1016/j.techfore.2018.04.016.

Prechelt, L. (2012). Early stopping - But when? *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7700*, 53–67, https://doi.org/10.1007/978-3-642-35289-8-5.

Probert, D. R., Farrukh, C. J. P., & Phaal, R. (2003). Technology roadmapping - Developing a practical approach for linking resources to strategic goals. *Proc. Instn Mech. Engrs*, *217*, 1183–1195 http://journals.sagepub.com/doi/pdf/10.1243/095440503322420115.

Provost, F. (2000). Machine learning from imbalanced data sets 101. *Proceedings of the AAAI 2000 Workshop*, https://doi.org/10.1.1.33.507.

Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural Networks*, *12*(1), 145–151, https://doi.org/10.1016/S0893-6080(98)00116-6.

Ranjan, C. (2019). Understanding Dropout with the Simplified Math behind it https://towardsdatascience.com/simplified-math-behind-dropout-in-deep-learning-6d50f3f47275.

Raturi, M. K., Sahoo, P. K., Mukherjee, S., & Tiwari, A. K. (2010). *Patinformatics – An Emerging Scientific Discipline* http://www.ssrn.com/abstract=1566067.

Rehurek, R. & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (pp. 45–50).: ELRA http://is.muni.cz/publication/884893/en.

Reitzig, M. (2003). What determines patent value?: Insights from the semiconductor industry. *Research Policy*, *32*(1), 13–26, https://doi.org/10.1016/S0048-7333(01)00193-7.

Reitzig, M. (2004). Improving patent valuations for management purposes - Validating new indicators by analyzing application rationales. *Research Policy*, *33*(6-7), 939–957, https://doi.org/10.1016/j.respol.2004.02.004.

Riedl, C., et al. (2016). Detecting figures and part labels in patents: competition-based development of graphics recognition algorithms. *International Journal on Document Analysis and Recognition*, *19*(2), 155–172, https://doi.org/10.1007/s10032-016-0260-8.

Robson, C. (2011). *Real world research*. Blackwell Publishing Malden, 4th edition.

Rocca, J. (2019). Ensemble methods: bagging, boosting and stacking. *Towards Data Science* https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205.

Rockett, K. (2010). Chapter 7 - Property Rights and Invention. In B. H. Hall & N. Rosenberg (Eds.), *Handbook of The Economics of Innovation, Vol. 1*, volume 1 of *Handbook of the Economics of Innovation* (pp. 315 – 380). North-Holland http://www.sciencedirect.com/science/article/pii/S0169721810010075.

Roelants, P. (2020). Softmax classification with cross-entropy. *Notes on Machine Learning* https://peterroelants.github.io/posts/cross-entropy-softmax/.

Rosenberg, N. (1994). *Exploring the Black Box : Technology, Economics, and History*. Cambridge University Press.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408, https://doi.org/10.1037/h0042519.

Ruder, S. (2017a). An overview of gradient descent optimization algorithms. *Blog sebastianruder - working paper article* http://arxiv.org/abs/1609.04747.

Ruder, S. (2017b). Transfer Learning - Machine Learning's Next Frontier. *Transfer Learning* https://ruder.io/transfer-learning/.

Russell, S. J. & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach* 4th Edition, Prentice Hall.

Sadowski, P. (2017). Notes on Backpropagation. *Department of Computer Science University of California Irvine*, *1*(4), 1–3.

Saltelli, A., et al. (2008). *Global Sensitivity Analysis: The Primer*. Wiley.

Saltelli, A., Tarantola, S., Campolongo, F., & Ratto, M. (2004). *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*. Wiley.

Samala, A. K. (2017). Hierarchical Softmax as output activation function in Neural Network https://becominghuman.ai/hierarchical-softmax-as-output-activation-function-in-neural-network-1d19089c4f49.

Sanjay, M. (2018). Why and how to Cross Validate a Model? *Towards Data Science* https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f.

Santurkar, S., Tsipras, D., Ilyas, A., & Madry, A. (2018). How does batch normalization help optimization? *Advances in Neural Information Processing Systems*, *2018-Decem*(NeurIPS), 2483–2493.

Sapsalis, E., van Pottelsberghe de la Potterie, B., & Navon, R. (2006). Academic versus industry patenting: An in-depth analysis of what determines patent value. *Research Policy*, *35*(10), 1631–1645, https://doi.org/10.1016/j.respol.2006.09.014.

Sargent, D. J. (2001). Comparison of artificial neural networks with other statistical approaches. *Cancer*, *91*(S8), 1636–1642, https://doi.org/10.1002/1097-0142(20010415)91:8+<1636::AID-CNCR1176>3.0.CO;2-D.

Saxena, S. (2018). Precision vs Recall. *Towards Data Science* https://towardsdatascience.com/precision-vs-recall-386cf9f89488.

Schakel, A. M. J. & Wilson, B. J. (2015). Measuring Word Significance using Distributed Representations of Words http://arxiv.org/abs/1508.02297.

Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Networks*, *61*, 85–117, https://doi.org/10.1016/j.neunet.2014.09.003.

Schneider, J. & Moore, A. W. (1997). A locally weighted learning tutorial. *cs.cmu.edu* https://www.cs.cmu.edu/~schneide/tut5/tut5.html.

Schraudolph, N. & Cummins, F. (2006). Introduction to Neural Networks. *CNL - Salk Institute* https://cnl.salk.edu/{~}schraudo/teach/NNcourse/intro.html.

Seif, G. (2018). Handling Imbalanced Datasets in Deep Learning. *Towards Data Science* https://towardsdatascience.com/handling-imbalanced-datasets-in-deep-learning-f48407a0e758.

Senapati, D. (2018). Grid Search vs Random Search. *Medium* https://medium.com/@senapati.dipak97/grid-search-vs-random-search-d34c92946318.

Seo, S., Lee, J. M., Yang, H., & Kim, S. (2019). Can AI tell emerging technologies: Evaluating the importance of quantitative features of technology. In *PICMET 2019 - Portland International Conference on Management of Engineering and Technology: Technology Management in the World of Intelligent Systems, Proceedings*.

Seru, A. (2014). Firm boundaries matter: Evidence from conglomerates and R&D activity. *Journal of Financial Economics*, *111*(2), 381–405, https://doi.org/10.1016/j.jfineco.2013.11.001.

Shaked, T., et al. (2016). Wide and deep machine learning models.

Shane, S. (2001). Technological opportunities and new firm creation. *Management Science*, *47*(2), 205–220, https://doi.org/10.1287/mnsc.47.2.205.9837.

Sharma, S. (2017a). Activation Functions in Neural Networks. *Towards Data Science* https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

Sharma, S. (2017b). What the Hell is Perceptron? - Towards Data Science. *Towards Data Science* https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53.

Shmueli, B. (2019). Multi-Class Metrics Made Simple, Part II: the F1-score. *Medium - Towards Data Science* https://towardsdatascience.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1.

Shubham, J. (2018). Ensemble Learning - Bagging and Boosting. *Medium - Becoming Human: Artificial Intelligence Magazine* https://becominghuman.ai/ensemble-learning-bagging-and-boosting-d20f38be9b1e.

Silva, R., Koshiyama, A., & Aristodemou, L. (2019). Linking Research Entities to Industrial Sectors: a hybrid methodology applied to Brazil's nanotechnology sector. In *Data for Policy Conference 2019, London, United Kingdom*.

Skripnikova, T. (2019). *Semantic Exploration of Text Documents with Multi-Faceted Metadata Employing Word Embeddings: The Patent Landscaping Use Case*. Technical report.

Smith, L. N. (2017). Cyclical learning rates for training neural networks. *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*, (April), 464–472, https://doi.org/10.1109/WACV.2017.58.

Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. (pp. 1–21). http://arxiv.org/abs/1803.09820.

Smith, L. N. & Topin, N. (2019). Super-convergence: very fast training of neural networks using large learning rates. (pp.3̃6)., https://doi.org/10.1117/12.2520589.

So-Young, S., Moon, H. T., Kim, S., & Kim, W. D. (2007). Method of technology valuation https://patents.google.com/patent/WO2007073063A1/en?oq=WO+2007{%}2F073063A1.

Soenksen, L. R. L. & Yazdi, Y. (2016). Stage-gate process for life sciences and medical innovation investment. *Technovation*, *62-63*(August 2016), 14–21, https://doi.org/10.1016/j.technovation.2017.03.003.

Sokolova, M. & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, *45*(4), 427–437, https://doi.org/10.1016/j.ipm.2009.03.002.

Squicciarini, M., Dernis, H., & Criscuolo, C. (2013). Measuring Patent Quality: Indicators of Technological and Economic Value. *OECD Science, Technology and Industry Working Papers*, (03), 70, https://doi.org/10.1787/5k4522wkw1r8-en.

Sreekumaran Nair, S., Mathew, M., & Nag, D. (2012). Effect of firm variables on patent price. *IIMB Management Review*, *24*(1), 40–47, https://doi.org/10.1016/j.iimb.2011.12.004.

Srivastava, N., Hinton, G., Sutskever, I., & Salakhutdinov, R. R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958, https://doi.org/10.5555/2627435.

Stack Overflow (2016). Ordering of batch normalization and dropout? *Stack Overflow* https://stackoverflow.com/questions/39691902/ordering-of-batch-normalization-and-dropout.

Stathakis, D. (2009). How many hidden layers and nodes? *International Journal of Remote Sensing*, *30*(8), 2133–2147, https://doi.org/10.1080/01431160802549278.

Stein, R. A., Jaques, P. A., Valiati, J. F., Alan, R., Jaques, P. A., & Francisco, J. (2019). An analysis of hierarchical text classification using word embeddings. *Information Sciences*, *471*, 216–232, https://doi.org/10.1016/j.ins.2018.09.001.

Striukova, L. (2007). Patents and corporate value creation: Theoretical approach. *Journal of Intellectual Capital*, *8*(3), 431–443, https://doi.org/10.1108/14691930710774858.

Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions.

Suh, J. H. (2015). Exploring the effect of structural patent indicators in forward patent citation networks on patent price from firm market value. *Technology Analysis & Strategic Management*, (March 2015), 1–18, https://doi.org/10.1080/09537325.2015.1011613.

Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, *23*(4), 687–719, https://doi.org/10.1142/S0218001409007326.

Sung, H. Y., Yeh, H. Y., Lin, J. K., & Chen, S. H. (2017). A visualization tool of patent topic evolution using a growing cell structure neural network. *Scientometrics*, *111*(3), 1267–1285, https://doi.org/10.1007/s11192-017-2361-7.

Suominen, A., Toivanen, H., & Seppänen, M. (2017). Firms' knowledge profiles: Mapping patent data with unsupervised learning. *Technological Forecasting and Social Change*, *115*, 131–142, https://doi.org/10.1016/j.techfore.2016.09.028.

Suzuki, J. (2011). Structural modeling of the value of patent. *Research Policy*, *40*(7), 986–1000, https://doi.org/10.1016/j.respol.2011.05.006.

Tahmooresnejad, L. & Beaudry, C. (2018). Do patents of academic funded researchers enjoy a longer life? A study of patent renewal decisions. *PLoS ONE*, *13*(8), 1–22, https://doi.org/10.1371/journal.pone.0202643.

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: An analysis and review. *International Journal of Forecasting*, *16*(4), 437–450, https://doi.org/10.1016/S0169-2070(00)00065-0.

Teece, D. J. (1986). Profiling from technological innovation: implications for integration, collaboration, licencing and public policy. *Research Policy*, *15*(February), 285–305, https://doi.org/10.1016/0048-7333(86)90027-2.

Tekic, Z. & Kukolj, D. (2013). Threat of litigation and patent value. *Research Technology Management*, *56*(2), 18–25, https://doi.org/10.5437/08956308X5602093.

Tenorio-González, A. C. & Morales, E. F. (2018). Automatic discovery of concepts and actions. *Expert Systems with Applications*, *92*, 192–205, https://doi.org/10.1016/j.eswa.2017.09.023.

Tewari, A. & Bartlett, P. L. (2007). On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, *8*, 1007–1025, https://doi.org/10.1007/11503415_10.

The British Library (2020). What is a patent https://www.bl.uk/business-and-ip-centre/articles/what-is-a-patent{#}.

The Pandas Development Team (2020). pandas-dev/pandas: Pandas https://doi.org/10.5281/zenodo.3509134.

Thoma, G. (2014). Composite value index of patent indicators: Factor analysis combining bibliographic and survey datasets. *World Patent Information*, *38*, 19–26, https://doi.org/10.1016/j.wpi.2014.05.005.

Thoma, G. (2016). *Patent management and valuation: The strategic and geographical dimension*. Taylor and Francis.

Thompson, M. J. (2017). The cost of patent protection: Renewal propensity. *World Patent Information*, *49*, 22–33, https://doi.org/10.1016/j.wpi.2017.02.002.

Thompson, P. (2010). Chapter 10 - Learning by Doing. In B. H. Hall & N. Rosenberg (Eds.), *Handbook of The Economics of Innovation, Vol. 1*, volume 1 of *Handbook of the Economics of Innovation* (pp. 429 – 476). North-Holland http://www.sciencedirect.com/science/article/pii/S0169721810010105.

Thorleuchter, D., den Poel, D. V., & Prinzie, A. (2010). A compared R&D-based and patent-based cross impact analysis for identifying relationships between technologies. *Technological Forecasting and Social Change*, *77*(7), 1037–1050, https://doi.org/10.1016/j.techfore.2010.03.002.

Tietze, F. (2012). *Technology market transactions: auctions, intermediaries and innovation*. Edward Elgar Publishing.

Tietze, F., Vimalnath, P., Aristodemou, L., & Molloy, J. (2020a). Crisis-Critical Intellectual Property : Findings from the COVID-19 Pandemic. *Centre for Technology Management working paper series*, (202004), 1–18, https://doi.org/10.17863/CAM.51142.

Tietze, F., Vimalnath, P., Aristodemou, L., & Molloy, J. (2020b). Crisis-Critical Intellectual Property: Findings From the COVID-19 Pandemic. *IEEE Transactions on Engineering Management*, (pp. 1–18)., https://doi.org/10.1109/TEM.2020.2996982.

Tobin, J. & Brainard, W. C. (1977). Asset Markets and the Cost of Capital. *Economic Progress Private Values and Public Policies: Essays in Honor of William Fellner*.

Trajtenberg, M. (1990). A penny for your quotes: patent citations and the value of innovations. *The RAND Journal of Economics*, *21*(1), 172–187, https://doi.org/10.2307/2555502.

Trajtenberg, M., Henderson, R., & Jaffe, A. (1997). *University Versus Corporate Patents: A Window On The Basicness Of Invention*, volume 5.

Trappey, A. J., Lupu, M., & Stjepandic, J. (2020a). Embrace artificial intelligence technologies for advanced analytics and management of intellectual properties. *World Patent Information*, *61*, 101970, https://doi.org/10.1016/j.wpi.2020.101970.

Trappey, A. J., Trappey, C. V., Govindarajan, U. H., & Sun, J. J. (2019). Patent Value Analysis Using Deep Learning Models - The Case of IoT Technology Mining for the Manufacturing Industry. *IEEE Transactions on Engineering Management*, *PP*, 1–13, https://doi.org/10.1109/TEM.2019.2957842.

Trappey, A. J., Trappey, C. V., Hareesh Govindarajan, U., Chuang, A. C., & Sun, J. J. (2017a). A review of essential standards and patent landscapes for the Internet of Things: A key enabler for Industry 4.0. *Advanced Engineering Informatics*, *33*, 208–229, https://doi.org/10.1016/j.aei.2016.11.007.

Trappey, A. J., Trappey, C. V., Wu, C. Y., & Lin, C. W. (2012). A patent quality analysis for innovative technology and product development. *Advanced Engineering Informatics*, *26*(1), 26–34, https://doi.org/10.1016/j.aei.2011.06.005.

Trappey, A. J., Trappey, C. V., Wu, J. L., & Wang, J. W. (2020b). Intelligent compilation of patent summaries using machine learning and natural language processing techniques. *Advanced Engineering Informatics*, *43*, https://doi.org/10.1016/j.aei.2019.101027.

Trappey, A. J. C., Hsu, F. C., Trappey, C. V., & Lin, C. I. (2006). Development of a patent document classification and search platform using a back-propagation network. *Expert Systems with Applications*, *31*(4), 755–765, https://doi.org/10.1016/j.eswa.2006.01.013.

Trappey, A. J. C., Trappey, C. V., & Lee, K. L. C. (2017b). Tracing the Evolution of Biomedical 3D Printing Technology Using Ontology-Based Patent Concept Analysis. *Technology Analysis & Strategic Management*, *29*(4), 339–352, https://doi.org/10.1080/09537325.2016.1211267.

Trappey, A. J. C., Trappey, C. V., Wu, C. Y., Fan, C. Y., & Lin, Y. L. (2013). Intelligent patent recommendation system for innovative design collaboration. *Journal of Network and Computer Applications*, *36*(6), 1441–1450, https://doi.org/10.1016/j.jnca.2013.02.035.

Trippe, A. (2015). *Guidelines for Preparing Patent Landscape Reports*. Technical report, World Intellectual Property Organisation http://www.wipo.int/edocs/pubdocs/en/wipo_pub_946.pdf.

Trippe, A. J. (2003). Patinformatics: Tasks to tools. *World Patent Information*, *25*(3), 211–221, https://doi.org/10.1016/S0172-2190(03)00079-6.

Turban, E., Aronson, J. E., Liang, T.-P., & Turban, E. (2005). *Decision Support Systems and Intelligent Systems*. Prentice-Hall, 7th edition http://sutlib2.sut.ac.th/sut{_}contents/H86360.pdf.

Uhm, D., Ryu, J. B., & Jun, S. (2020). Patent data analysis of artificial intelligence using bayesian interval estimation. *Applied Sciences (Switzerland)*, *10*(2), https://doi.org/10.3390/app10020570.

USPTO (2020). Maintain your patent https://www.uspto.gov/patents-maintaining-patent/maintain-your-patent.

Valchanov, I. (2018). False Positive and False Negative https://towardsdatascience.com/false-positive-and-false-negative-b29df2c60aca.

Van Der Maaten, L. (2009). Learning a parametric embedding by preserving local structure. *Journal of Machine Learning Research*, *5*, 384–391.

Van Der Maaten, L. (2010). Accelerating t-SNE using Tree-Based Algorithms Laurens. *Journal of Machine Learning Research*, *15*, 1–21.

Van Der Maaten, L. (2020). t-SNE. *t-SNE Implementations and Examples* https://lvdmaaten.github.io/tsne/.

Van Der Maaten, L. & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605, https://doi.org/10.1007/s10479-011-0841-3.

Van Rossum, G. & de Boer, J. (1991). Interactively Testing Remote Servers Using the Python Programming Language. *CWI Quarterly*.

Van Rossum, G. & Drake, F. L. (1995). Python Reference Manual. *Centrum voor Wiskunde en Informatica Amsterdam*.

Van Rossum, G. & Drake, F. L. (2009). Python 3 Reference Manual. *CreateSpace*.

Van Veen, F. (2016). The Neural Network Zoo. *The Asimov institute blog* https://www.asimovinstitute.org/neural-network-zoo/.

van Wieringen, W. N. (2018). *Lecture notes on ridge regression*. Technical report, VU University Asterdam http://arxiv.org/abs/1509.09169.

van Zeebroeck, N. (2011). The puzzle of patent value indicators. *Economics of Innovation and New Technology*, *20*(1), 33–62, https://doi.org/10.1080/10438590903038256.

van Zeebroeck, N. & van Pottelsberghe de la Potterie, B. (2011a). Filing strategies and patent value. *Economics of Innovation and New Technology*, *20*(6), 539–561, https://doi.org/10.1080/10438591003668646.

van Zeebroeck, N. & van Pottelsberghe de la Potterie, B. (2011b). The vulnerability of patent value determinants. *Economics of Innovation and New Technology*, *20*(3), 283–308, https://doi.org/10.1080/10438591003668638.

Venugopalan, S. & Rai, V. (2015). Topic based classification and pattern identification in patents. *Technological Forecasting and Social Change*, *94*, 236–250, https://doi.org/10.1016/j.techfore.2014.10.006.

Verbano, C. & Nosella, a. (2010). Addressing R&D investment decisions: A cross analysis of R&D project selection methods. *European Journal of Innovation Management*, *13*(3), 355–379, https://doi.org/10.1108/14601061011060166.

Verhoeven, D., Bakker, J., & Veugelers, R. (2016). Measuring technological novelty with patent-based indicators. *Research Policy*, *45*(3), 707–723, https://doi.org/10.1016/j.respol.2015.11.010.

Veugelers, R. & Wang, J. (2019). Scientific novelty and technological impact. *Research Policy*, *48*(6), 1362–1372, https://doi.org/10.1016/j.respol.2019.01.019.

Vrochidis, S., Moumtzidou, A., & Kompatsiaris, I. (2012). Concept-based patent image retrieval. *World Patent Information*, *34*(4), 292–303, https://doi.org/10.1016/j.wpi.2012.07.002.

Wagner, S. & Wakeman, S. (2016). What do patent-based measures tell us about product commercialization? Evidence from the pharmaceutical industry. *Research Policy*, *45*(5), 1091–1102, https://doi.org/10.1016/j.respol.2016.02.006.

Wang, J. & Chen, Y. J. (2019). A novelty detection patent mining approach for analyzing technological opportunities. *Advanced Engineering Informatics*, *42*(August 2018), 100941, https://doi.org/10.1016/j.aei.2019.100941.

Wang, K. (2015). Field Study of Patent Strategies from Patent Map on Big Data : An Empirical Case of Big Data Application Platform in Taiwan. In *ITRI studio of patent factory*.

Wang, S., et al. (2019). Massive computational acceleration by using neural networks to emulate mechanism-based biological models. *Nature Communications*, *10*(1), 1–9, https://doi.org/10.1038/s41467-019-12342-y.

Wattenberg, M., Viégas, F., & Johnson, I. (2016). How to Use t-SNE Effectively. *Distill*, *1*(10), https://doi.org/10.23915/distill.00002.

Weber, M., Fürst, M., & Zöllner, J. M. (2019). Automated Focal Loss for Image based Object Detection. *Working paper article* http://arxiv.org/abs/1904.09048.

Wei, C. (2019). Multi-class classification with focal loss for imbalanced datasets. *DLology* https://www.dlology.com/blog/multi-class-classification-with-focal-loss-for-imbalanced-datasets/.

Whitehead, M., Johnson, D. K. N., & Whitehead, M. (2017). A tool for visualizing and exploring relationships among cancer-related patents. *FLAIRS 2017 - Proceedings of the 30th International Florida Artificial Intelligence Research Society Conference*, (pp. 235–238). http://cs.coloradocollege.edu/~mwhitehead/CancerMoonshot/documents/iaai.pdf.

Whittington, J. C. & Bogacz, R. (2019). Theories of Error Back-Propagation in the Brain. *Trends in Cognitive Sciences*, *23*(3), 235–250, https://doi.org/10.1016/j.tics.2018.12.005.

Widrow, B. & Lehr, M. A. (1993). Adaptive Neural Networks and their Applications. *International Journal of Intelligent Systems*, *8*, 453–507.

WIPO (2004). *WIPO Intellectual Property Handbook: Policy, Law and Use*. Technical report, World Intellectual Property Organization (WIPO), Geneva.

WIPO (2009). *IPC and Technology Concordance Table*. Technical report, World Intellectial Property Organisation (WIPO), Geneva https://www.wipo.int/meetings/en/doc_details.jsp?doc_id=117672.

WIPO (2010). *WIPO Patent Drafting Manual*. Technical report, World Intellectual Property Organization (WIPO), Geneva http://www.wipo.int/edocs/pubdocs/en/patents/867/wipo_pub_867.pdf.

WIPO (2019a). WIPO IP Facts and Figures 2019. *WIPO publications* https://www.wipo.int/edocs/pubdocs/en/wipo_pub_943_2019.pdf.

WIPO (2019b). *WIPO Technology Trends 2019: Artificial Intelligence*. Technical report, World Intellectual Property Organisation (WIPO), Geneva.

WIPO (2019c). *World Intellectual Property Indicators 2019*. Technical report, World Intellectual Property Organization (WIPO), Geneva.

WIPO (2020). Patents https://www.wipo.int/patents/en/.

Woo, H.-G., Yeom, J., & Lee, C. (2019). Screening early stage ideas in technology development processes: a text mining and k -nearest neighbours approach using patent information. *Technology Analysis & Strategic Management*, *31*(5), 532–545, https://doi.org/10.1080/09537325.2018.1523386.

Wu, J. L., Chang, P. C., Tsao, C. C., & Fan, C. Y. (2016). A patent quality analysis and classification system using self-organizing maps with support vector machine. *Applied Soft Computing Journal*, *41*, 305–316, https://doi.org/10.1016/j.asoc.2016.01.020.

Wu, J.-l. L. (2019). Patent Quality Classification System Using the Feature Extractor of Deep Recurrent Neural Network. In *2019 IEEE International Conference on Big Data and Smart Computing, BigComp 2019 - Proceedings*.

Wu, W., Maier, H. R., Dandy, G. C., & May, R. (2012a). Exploring the Impact of Data Splitting Methods on Artificial Neural Network Models. In *10th International conference on Hydroinformatics, Hamburg, Germany*.

Wu, W., May, R., Dandy, G. C., & Maier, H. R. (2012b). A method for comparing data splitting approaches for developing hydrological (ANN) models. *2012 International Congress on Environmental Modelling and Software. Managing Resources of a Limited Planet: Pathways and Visions under Uncertainty, Sixth Biennial Meeting*, (pp. 1620–1627).

Wu, Y. & Radewagen, R. (2017). 7 Techniques to Handle Imbalanced Data https://www.kdnuggets.com/2017/06/7-techniques-handle-imbalanced-data.html.

Xu, Y. & Goodacre, R. (2018). On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *Journal of Analysis and Testing*, *2*(3), 249–262, https://doi.org/10.1007/s41664-018-0068-2.

Yin, Z. & Shen, Y. (2018). On the Dimensionality of Word Embeddings. In *32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montreal, Canada.*

Zhai, Z., et al. (2019). Improving Chemical Named Entity Recognition in Patents with Contextualized Word Embeddings. *Working paper article*, (pp. 328–338)., https://doi.org/10.18653/v1/w19-5035.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2016a). Understanding deep learning requires rethinking generalization. *Google Brain working paper series*, https://doi.org/10.1109/TKDE.2015.2507132.

Zhang, G., Duan, H., Wang, S., & Zhang, Q. (2018). Comparative technological advantages between China and developed areas in respect of energy production: Quantitative and qualitative measurements based on patents. *Energy*, *162*, 1223–1233, https://doi.org/10.1016/j.energy.2018.08.081.

Zhang, L.-W., Zhang, Q., Wang, X.-F., & Zhu, D.-H. (2009). Application research of robust LS-SVM regression model in forecasting patent application counts. *Journal of Beijing Institute of Technology (English Edition)*, *18*(4).

Zhang, S., Yuan, C.-c. C., Chang, K.-c. C., & Ken, Y. (2012). Exploring the nonlinear effects of patent H index , patent citations , and essential technological strength on corporate performance by using artificial neural network. *Journal of Informetrics*, *6*(4), 485–495, https://doi.org/10.1016/j.joi.2012.03.006.

Zhang, X. (2014). Interactive patent classification based on multi-classifier fusion and active learning. *Neurocomputing*, *127*, 200–205, https://doi.org/10.1016/j.neucom.2013.08.013.

Zhang, Y., Qian, Y., Huang, Y., Guo, Y., Zhang, G., & Lu, J. (2017). An entropy-based indicator system for measuring the potential of patents in technological innovation: rejecting moderation. *Scientometrics*, *111*(3), 1925–1946, https://doi.org/10.1007/s11192-017-2337-7.

Zhang, Y., Xu, J., Chen, H., Wang, J., Wu, Y., Prakasam, M., & Xu, H. (2016b). Chemical named entity recognition in patents by domain knowledge and unsupervised feature learning. *Database*, *2016*(December 2017), 1–10, https://doi.org/10.1093/database/baw049.

Zhou, X. & Lin, H. (2008). *Sensitivity Analysis*, (pp. 1046–1048). Springer US: Boston, MA https://doi.org/10.1007/978-0-387-35973-1_1191.

Zhu, F., Wang, X., Zhu, D., & Liu, Y. (2015). A Supervised Requirement-oriented Patent Classification Scheme Based on the Combination of Metadata and Citation Information. *International Journal of Computational Intelligence Systems*, *8*(3), 502–516, https://doi.org/10.1080/18756891.2015.1023588.

Zou, Q., Xie, S., Lin, Z., Wu, M., & Ju, Y. (2016). Finding the Best Classification Threshold in Imbalanced Classification. *Big Data Research*, *5*, 2–8, https://doi.org/10.1016/j.bdr.2015.12.001.