The Use of AI Techniques to Determine the Legitimacy of Patents' Originality

MSc Research Project
Cloud Computing

Victoria Ekefre
Student ID: 21213771

School of Computing
National College of Ireland

Supervisor: Rashid Mijumbi

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Victoria Ekefre
Student ID:	21213771
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Rashid Mijumbi
Submission Due Date:	14/08/2023
Project Title:	The Use of AI Techniques to Determine the Legitimacy of Patents' Originality
Word Count:	XXX
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	17th September 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

The Use of AI Techniques to Determine the Legitimacy of Patents' Originality

Victoria Ekefre
21213771

Abstract

To ensure that only cutting-edge concepts are patentable, it is crucial in intellectual property law to evaluate a patent's originality. However, the procedure is frequently time-consuming, expensive, and prone to prejudice. In order to address the difficulties of patent analysis and establish the veracity of patent originality, this research focuses on employing artificial intelligence (AI) tools. The validity of patent claims, descriptions, and illustrations has previously been evaluated using AI-based systems that include cutting-edge machine learning, natural language processing, and image processing techniques. These systems lack the ability to process complex legal language, detect variations, and analyze intricate diagrams using sophisticated graph neural network techniques. In this study, a graph neural network model is utilized to extract significant data and separate real patents from counterfeit ones by making predictions based on learned parameters. The implemented model achieved an impressive prediction accuracy of 1.0, demonstrating its efficacy in distinguishing genuine patents from counterfeit ones.

Keywords: patient originality, AI-based systems, graph neural networks, learned parameters, prediction accuracy.

1 Introduction

For the growth and protection of new ideas as well as for fair recompense for inventors, intellectual property protection, particularly through patents, is crucial Higham et al. (2021). It might be challenging to judge a patent's novelty and validity, though. In light of recent advancements in this area, researchers are looking into how artificial intelligence (AI) might enhance the patent inspection process Jobin et al. (2019). This study seeks to analyze the present state of research on AI applications for determining patent originality in order to build a more reliable and accurate approach. Due to the growing volume of patent applications, it is increasingly impractical for patent examiners to manually evaluate patent applications for novelty and inventiveness Lim (2018). Therefore, automating this procedure with AI approaches is particularly effective in order to save money and time Liu et al. (2011).

The use of AI in patent assessment is a crucial area of research because of its potential effects on innovation, intellectual property rights, and legal difficulties Gervais (2019). Natural language processing (NLP), machine learning (ML), and deep learning (DL) approaches have been the focus of recent research that tries to analyse and condense patent texts, identify prior art, and evaluate the quality and validity of patent claims Trappey et al. (2020). They built an intelligent system that uses NLP and ML to construct patent summaries. Sokhansanj and Rosen (2022) used DL to forecast results in inter-patent review procedures at the US Patent Trial and Appeal Board. To expedite prior art searches and make it simpler to locate new patents, further research has concentrated on text analysis and machine learning Setchi and Spasic (2020). Despite the promising outcomes of prior research on AI, obtaining consistent results depends on the correctness of the algorithms and the availability of high-quality data Cheng et al. (2022). The goal of this study is to increase the accuracy with which AI tools can correctly predict patent originality and this will be achieved by applying a graph neural network model. The model will extract important information and use it to distinguish between real patents from fake ones. The GNN's capacity to recognize intricate links and patterns in patent papers enhances predictions of patent legitimacy. The improved accuracy could help researchers, policymakers, and patent examiners by increasing the effectiveness and efficiency of the patent examination procedures, protecting real innovations, and spotting fake patents.

1.1 Research Problem

Can the authenticity of a patent be determined using artificial intelligence methods, such as natural language processing and machine learning algorithms, by extracting and classifying relevant data from collections of patent documents and identifying signs of genuine innovation as opposed to fakes?

1.2 Proposed Solution

The project will create a new algorithm using GNN to effectively extract and classify significant data from patent repositories in order to distinguish real patents from fake ones. The goal of this is to attempt to increase the accuracy of AI-based prediction.

The scientific community stands to gain from the suggested method's ability to improve intellectual property protection and raise the bar for patent review. In addition, by conducting this study, the time and resources devoted to researching counterfeit patents may be reduced, allowing for the prioritization of other crucial actions.

The introduction provides an overview of the study, identifies the research problem, and suggests a solution. The section that follows conducts a comprehensive analysis of the extant literature on AI in patent evaluation, analyzing the benefits and drawbacks of AI methodologies. The following section describes the proposed methodology for conducting research, including data collection, manipulation, and analysis. The implementation section concentrates on the pragmatic application of GNN, to differentiate between authentic and counterfeit patents in order to automate the patent examination workflow. Afterward, an evaluation of the outcomes attained through the implementation of AI methodologies is provided, addressing the accuracy and dependability of the employed algorithms and assessing the quality of the utilized data. This exhaustive study concludes by highlighting the potential of GNN to improve the accuracy of patent evaluation. This study emphasizes the significance of incorporating AI into patent evaluation processes by conducting a thorough literature review, proposing an efficient methodology, implementing AI-driven techniques, and analyzing the results.

2 Related Work

For the protection of intellectual property rights and the encouragement of innovation, it is essential to determine the originality and validity of patents. Researchers have long studied the capacity to predict patent originality, and recent advancements in artificial intelligence (AI) techniques have produced encouraging results in this area.

2.1 Before The Application Of AI

Prior to the development of AI, academics mostly used conventional metrics and indicators to assess the novelty and originality of patents.Verhoeven et al. (2016) suggested using patent-based indicators as a way to assess technological originality. To evaluate the degree of uniqueness and the impact of a patent within its technological sector, these indicators covered a variety of elements, such as citations, forward citations, and backward citations. Similar to this, Harrigan et al. (2016) used a distance metric to operationalize patent originality. This measurement attempted to put a number on the uniqueness of a patent by measuring how different it is from earlier works. Despite the fact that these conventional methods provided insightful analyses of patent innovation and originality, they were inherently incapable of capturing the complex linkages and patterns seen in patent papers. Patents are intricate legal documents that frequently include lengthy technical details, legalese, and prior art citations. It was difficult to extract useful information from such documents using only indicators and distance measurements. The semantic significance, context, and interconnectedness of the information contained in the patents were difficult for conventional techniques to take into consideration. The evaluation of patent uniqueness and originality was thereby constrained to a small number of quantitative variables and crude measures of dissimilarity. These techniques could ignore crucial details that are crucial in assessing the genuine originality of a patent, such as conceptual innovations, inventive steps, or innovative combinations of already existing technologies. It became clear that more sophisticated methods were required to accurately capture the intricate linkages, contextual data, and semantic nuances found in patent filings. Dou et al. (2005) presented a groundbreaking study that established the groundwork for the strategic application of patent analysis. Their investigation focused on patent analysis as a means of competitive technical intelligence and creative thought. Despite the fact that AI techniques were not prevalent at this time, the authors emphasized the importance of patent analysis for comprehending technological advancements and competitor strategies. This early research demonstrated the utility of patent analysis as a tool for both researchers and inventors. Liang et al. (2008) investigated the application of text mining techniques to patent analysis, concentrating on the TRIZ (Theory of Inventive Problem Solving) context. Using text mining, the researchers sought to identify patterns and essential information within patent texts, thereby facilitating the analysis of TRIZ principles applied to patents. This study demonstrated the potential of sophisticated analytical techniques to extract valuable knowledge from patent texts, despite not being explicitly AI-focused.Karvonen and Kässi (2011) delved deeper into the function of patent analysis, investigating its capacity for technological convergence analysis. The authors used co-word analysis and network visualization techniques to disclose connections and overlaps between various technological fields as AI methods gained traction. Their results demonstrated the burgeoning potential of AI-driven patent analysis for identifying emergent technologies and evaluating their impact on technological convergence.

2.2 Application Of AI-based Systems

AI-based systems have gained traction in this field and have shown promising results in prediction. The increasing volume of patent applications and the need for thorough prior art investigations are just two of the challenges faced by patent examiners Hegde and Hegde (2022). Artificial intelligence (AI) techniques like machine learning and natural language processing are being proposed as potential solutions to automate and enhance the patent examination process and ensure reliable assessments of patent novelty.Zhou et al. (2021) provided a deep learning method to find novel innovations in patents that stand out from the crowd. The authors used convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to automatically find and classify outlier patents that showed the presence of developing technologies in the CNC machine tool market. This study demonstrated how AI techniques can be used to track new technologies and identify innovations with the potential to change the game. InnoVAE, a generative AI model, was created specifically for understanding patents and innovation Cheng et al. (2022). In order to extract latent representations from patent texts and identify hidden patterns, the researchers used sophisticated generative models like variational autoencoders (VAEs). This approach made it easier to understand patents in greater depth, which ultimately promotes the development of new technologies and innovation in a variety of industries. Another work by You et al. (2021) focuses on evaluating the quality of patents using deep learning techniques. The value of teamwork in training is underlined, and using several models to enhance evaluation is also emphasized. By utilizing the capabilities of deep neural networks, researchers showed the potential of artificial intelligence (AI) in assessing patent applications for novelty, inventiveness, and technical contribution, resulting in more accurate and dependable findings. Furthermore, using machine learning and natural language processing methods,Trappey et al. (2020) proposed an intelligent system that can construct patent summaries. The study illustrated how artificial intelligence (AI) might examine patent writings in the future and extract significant information from them. Thanks to the use of sophisticated mathematical procedures, the system was able to increase the efficiency of patent inspection and aid in the localization of patents that may have been violated. To extract meaningful information from patent documents, a cutting-edge framework that incorporates AI approaches to analyze patents, such as natural language processing, machine learning, and data mining, was utilized Son et al. (2022). This system demonstrated how AI may help uncover pertinent prior art, speed up patent analysis, and enhance patent inspection. Searching for prior art patents can also be done using text analysis and machine learning techniques Setchi and Spasic (2020). The speed with which prior art is found is a key factor in judging how original a patent is. Alderucci and Ashley (2020) also performed a literature review in which they looked into the use of AI to analyze the issue of patent claim indefiniteness. Their primary focus was on using AI techniques to assess the precision and completeness of patent claims. They exposed instances of unclear language and conflicting specifications inside patent claims using AI-driven analysis to shed light on concerns about patent originality through an examination of patent claims. Higham et al. (2021) provided a methodical approach for examining and quantifying patent quality. The authors stressed the significance of using impartial measures and indicators to assess the legitimacy and calibre of patents. The researchers used artificial intelligence methods, including machine learning and natural language processing, in order to build reliable models and procedures that permit a thorough assessment of the quality of patents. By using various artificial intelligence techniques, such as data analysis and clustering algorithms, it is possible to estimate the distance between assets and determine the degree of intellectual property assets' uniqueness Ragot (2022). This quantitative approach can be used to assess the uniqueness and originality of patents. The method put forth offers helpful insights into how to gauge a patent's originality and has the ability to raise the objectivity and uniformity of evaluation standards in this field.

Choi et al. (2022) investigated the use of deep learning techniques, such as transformer architectures and graph embeddings, for patent landscaping. The authors proposed a methodology that employs transformer models to extract meaningful information from patent texts, thereby enabling exhaustive patent analysis. In addition, the incorporation of graph embeddings permits the incorporation of structural information from patent citation networks, thereby enhancing the model's predictive capabilities.

2.3 Real Life Application Of AI Approaches

According to studies, big data analytics, AI, and machine learning are having an impact on intellectual property law, particularly in regard to data ownership, privacy, and copyright protection Gervais (2019). This highlights the critical need for adaptable legal frameworks that can keep up with the big data and AI industries' rapid technological changes. In the context of innovation and creativity, Lim (2018) investigated the connection between intellectual property (IP) and artificial intelligence (AI). The report emphasized how AI has had a significant impact on many areas of intellectual property law, including trademark use, copyright protection, and patent evaluation. Additionally, he looked at the advantages and difficulties of integrating AI into the field of intellectual property, highlighting the necessity of regulatory frameworks that successfully strike a balance between innovation and legal issues. Sokhansanj and Rosen (2022) predicted the outcomes of IPR cases using deep learning techniques. Their findings made it clear how crucial artificial intelligence (AI) is to enhance the decision-making process in the patent system. The study used cutting-edge deep learning models to analyze the preliminary response briefs of patent owners and show their capacity to forecast the results of IPR procedures. This offers insightful information about the patent examination process as well as aids examiners in making better-informed conclusions. AI approaches have also been applied in strategic decision-making in the AI industry via computational models and data analysis techniques on patents Li et al. (2023). This research was made to help industries become more competitive in innovation. Artificial intelligence techniques were used to collect patent data, which was then used as a roadmap for choosing important innovations. Patent map analysis was used to uncover promising smart farm technologies and develop a technology roadmap Chun et al. (2021). This research identified significant technology clusters and potential development paths through the analysis of patent data and the display of technological linkages.

A graph neural network model was used to forecast the stock market. GNN's ability to capture intricate relationships within patent citation networks and the correlation between these relationships and stock market trends was used in this application. The authors modelled the complex interconnections between patents using graph-based representations of patent citations, allowing for more accurate predictions of core patents. By incorporating stock market data, they demonstrate that their model has practical applications beyond conventional patent analysis Han et al. (2022).

A method for extracting keywords from patents was given by Kim et al. (2018) in an effort to further the practice of sustainable technology management. They examined patent texts and extracted pertinent keywords in order to identify sustainable technologies and evaluate their impacts on the environment and society. This study illustrated how natural language processing techniques in artificial intelligence can be utilized to simplify the identification and administration of eco-friendly technologies. It can be difficult to evaluate early-stage ideas in technology development processes, but the authors of a recent study Woo et al. (2019) introduced a novel method that combines text mining with the k-nearest neighbours (k-NN) algorithm to evaluate patent information. This technique aims to find unique technological concepts by examining patent texts for significant data. This study demonstrated how the use of AI techniques in conjunction with patent data has the potential to significantly enhance technology scouting and innovation management.

2.4 Summary

Intricate links and patterns in patent papers were difficult for traditional approaches for evaluating patent originality. Artificial intelligence-based methods that use machine learning, natural language processing, and deep learning have become more popular for determining the quality and originality of patents. These systems analyse claim indefiniteness, construct summaries, evaluate novelty and inventiveness, find hidden patterns, extract pertinent information from patent texts, and evaluate novelty and inventiveness. AI techniques have found practical uses in intellectual property, influencing copyright protection, data ownership, and the law. They have also been used to forecast IPR case outcomes, support strategic decision-making, locate sustainable innovations, assess patent data for technology scouting, and streamline the search for and management of environmentally friendly technologies.

While AI-based methods have already shown potential in predicting patent originality, Graph Neural Networks (GNNs) integration can improve accuracy even more. GNNs are created expressly to capture the intricate connections and linkages seen in patent papers. GNNs are able to learn from these graph structures and take into account both local and global information by modeling patent documents as graphs, with nodes denoting various elements and edges denoting relationships. This makes it possible for GNNs to find complex patterns, locate pertinent prior art, and perform a more thorough assessment of the novelty and inventiveness of patents.

3 Methodology

This project tends to improve the accuracy of previous AI-based solutions for the detection of patent originality which will be achieved through the application of a new cutting-edge technology, 'Graph Neural Network (GNN).'

3.1 Importance of GNN Model

GNNs are designed to handle graph-structured data, capturing dependencies and interactions among patent network entities. They can handle heterogeneous data by incorporating different types of node and edge features, leveraging multiple sources of data for prediction. GNNs can learn from local and global patterns in the graph structure, allowing nodes to acquire knowledge from their neighbours and aggregate data from the entire graph. This property is advantageous for predicting patent validity, as it evaluates the influence of connected entities and identifies significant patterns contributing to a patent's validity or invalidity. GNNs also demonstrate excellent scalability and efficacy in managing large-scale graphs, making them suitable for patent data analysis.

3.2 Approach

Graph Neural Networks (GNNs) are models of deep learning that process and analyse graph-structured data. Their popularity is attributable to their capacity to manage complex relationships and dependencies in graph-structured data. Graph Convolutional Networks (GCNs), GraphSAGE, Graph Isomorphism Networks (GINs), and Graph Attention Networks (GATs) are all examples of GNNs. GCNs aggregate information from neighbours to update representations, but they may not effectively manage long-distance dependencies. GraphSAGE enhances scalability and captures global graph data, whereas GINs are permutation-invariant and perform well in both transductive and inductive learning environments. GAT is an efficient algorithm for dealing with long-range dependencies in complex graph relationships. It weights neighbouring node characteristics adaptively, focusing on pertinent nodes and capturing significant patterns. GATs excel at node classification, link prediction, and graph classification, and are appropriate for large-scale real-world applications. Their attention coefficients provide interpretability by indicating how neighboring nodes affect the representation of each node.

GAT is a perfect fit for this research, and the following are methods we intend to apply in the course of this research:

3.2.1 Data Collection

The Google Patent Public Data is a publicly available dataset that can be accessed using the Kaggle platform (Abbas et al., 2014). Google Patent Public Data offers a collection of patent documents that is both extensive and varied. These documents cover a wide variety of business sectors as well as technological spheres. This dataset provides a great resource for our analysis, as it offers a vast amount of patent information that can be used to study and assess the originality of patents. The collection contains a tremendous amount of patents and the metadata that is connected with them, which includes patent titles, abstracts, inventors' information, assignees' information, and citation information. This amount of information makes it possible to conduct a comprehensive investigation of the originality of patents by investigating the connections between inventors, assignees, and citations.

3.2.2 Data Preprocessing

Data cleansing is a crucial stage in the data analysis process, addressing various data quality concerns. To address missing values, we will use machine learning and label encoders to check for missing data and outliers. Exploratory data analysis (EDA) will be performed using libraries like Matplotlib and Seaborn for data visualization. The describe() function will verify summary statistics of numerical columns, and the dtypes attribute will designate numerical, categorical, and textual data types. The dataset's shape will be displayed, and a correlation matrix will be generated. Finally, the updated dataset will be split into train, test, and validation sets, with 70% for the training set and 30% for the test and validation sets.

3.3 Graph Attention network

3.3.1 Graph Construction

Patents are the foundation of innovation and technological advancement, protecting concepts and ideas while encouraging further innovation. Understanding the connections among patents is crucial as their number keeps increasing continuously. The creation of patent citation network graphs is a useful strategy for gaining this understanding. In order to efficiently manipulate and traverse such graphs, I will use adjacency lists or adjacency matrices. It emphasises the importance of each step and shows how patent citation networks serve as the foundation for advanced analysis and artificial intelligence techniques. The key components needed for this project are a patent identifier, titles, abstracts, claims, inventor information, and citation references. Each patent must be represented as a node in order to convert the raw patent data into a coherent network. Each patent is given a special identity, or node ID, to make sure that every invention can be identified in the network. The patent identifier and a distinct number can be combined to create the node ID, ensuring the uniqueness of each node. To ensure that subsequent actions are as effective as possible, the patent citation network graph must be represented correctly, using adjacency matrices and adjacency lists. Each node (patent) in the adjacency list technique keeps a list of its neighbouring nodes, which stand in for the patents it cites. The citation relationships are effectively captured when the adjacency list is implemented as a dictionary or a collection of linked lists. Adjacency matrices, on the other hand, use a two-dimensional matrix to express the connections between nodes. Each cell in the matrix represents an edge between two patents, and it is possible to store extra attributes in these cells, such as the kind, year, or significance score. Adjacency matrices need more memory, but they offer constant time lookups for edge existence, facilitating quicker graph operations. The next stage is to build edges that capture the citation relationships present in the data once the nodes have been constructed. The connected web of intellectual influence is built by adding a directed edge from the citing patent to the referenced patent for each citation that is discovered.

The snippet in Figure 10 below is the pseudocode for the citation network graph

```
Step 1: Create a directed graph to represent the citation relationships.
citation_graph = CreateDirectedGraph()

Step 2: Add nodes (patents) to the graph.
citation_graph.AddNode(patent, validity=default_validity,
abstract=default_abstract, filing_date=default_filing_date,
x=default_filing_date)

Step 3: Update attributes with actual values where they exist.
for each row in df:

Step 4 (Optional): Add validity information as a node attribute.

Step 5 (Optional): Add other features as node attributes.

Step 6: Normalize 'filing_date' feature using StandardScaler.

Step 7: Convert citation_graph to a DataFrame for train-test split.

# Extract the node features (validity, abstract, filing_date) into separate
columns.

Step 8: Train-test split (80/20, random_state=42)

Step 9: Create separate subgraphs for the training and testing sets.
train_graph = citation_graph.Subgraph(train_nodes['patent'])
test_graph = citation_graph.Subgraph(test_nodes['patent'])
```

Figure 1: Pseudocode For Citation Network Graph

From the pseudocode snippet above, the following steps are implemented as detailed below:

Make a directed graph: A directed graph called a "citation graph" serves as a representation of the citation relationships between patents. Each node in this graph corresponds to a patent, and the directed edges show the citations between patents.

Add nodes to the graph: The citation graph gains nodes (patents). Each patent has default values for validity, abstract, filing date, and x (which is default filing date) when it is first created. Actual values for these attributes will be added later.

Update Node Attributes with their actual values: The algorithm loops through the data and updates the node attributes in the citation graph with actual values from the data frame for each row in the dataframe df. On the basis of the corresponding values in the DataFrame, the attributes validity, abstract, and filing date are updated for each patent node. The default values predetermined earlier are applied if any of these attributes are missing from the data frame.

Add Validity Information as a Node Attribute: 'Validity' information is added as a node attribute (validity) to the citation graph if the data frame df contains it. This step is optional and is dependent on the 'validity' data being present in the data frame.

Add Additional features as node attributes: Similar to this, 'abstract' and 'filing date' information are added as node attributes (abstract and filing date) to the citation graph if they are present in the DataFrame df. The 'abstract' attribute is calculated as the length of the 'abstract localized' text, and the 'filing date' is used as a feature for later normalization.

Filing date feature should be normalized using StandardScaler: The 'filing date' feature is taken from the node attributes and applied to each node in the citation graph. The StandardScaler from sklearn preprocessing is then used to normalise it. By ensuring that the scales of the feature values are comparable through normalization, learning is prevented from being dominated by any one feature.

Convert Graph to a data frame for the Train-Test Split: For the train-test split, the citation graph is transformed into a (nodes df). Each node (patent) and its attributes (validity, abstract, and normalized filing date) will be described in the DataFrame.

Using the train test split function from sklearn: For model selection, the dataframe nodes df is divided into training (train nodes) and testing (test nodes) sets. To ensure reproducibility, the random state is set to 42, and the split is carried out in an 80/20 ratio.

Separate Training and Testing Sets' Subgraphs: The citation graph is split into two distinct subgraphs, the train graph and the test graph. Only the nodes that are a part of the training set and testing set are included in these subgraphs, respectively. This procedure aids in separating the training and testing data for additional modeling or analysis.

3.3.2 Aggregation

The GAT model intelligently aggregates data using attention weights, thereby enhancing its comprehension of context and influence. This method enables a comprehensive comprehension of the patent citation network graph's intricate web of intellectual interactions. By combining data from nearby nodes, the model reveals complex patterns and trends that are not discernible when examining individual patents in isolation. This broader perspective enables a more thorough investigation of the patent citation network, thereby shedding light on the dissemination of ideas, technological advancement, and the influence of notable inventors.

3.3.3 Graph Attention Network Model Training

The refined node representations generated from the aggregation step—the first step in training the GAT model, are used to create the model architecture for the Graph Attention Network (GAT). The patent citation network graph's many linkages and interdependencies are captured by the GAT model's various layers, each of which has its own set of learnable weights and biases.

The attention mechanism is incorporated into each GAT layer to spread information across the graph. The attention mechanism, which distributes importance weights to neighbouring nodes during this propagation process, enables the GAT model to focus only on significant patents when assessing the novelty and importance of specific patents. The model's depiction of the patent citation network is further enhanced by the use of numerous GAT layers, which enable it to capture relationships of a progressively higher degree.

The architecture of the GAT model may be changed to fit the particular needs of this research, for example, by changing the number of layers or attention heads. With these adjustments, researchers can improve the model's performance and adapt it to the specifics of the patent citation data.

The following stage entails developing suitable loss functions to measure the discrepancy between the training data labels and the expected labels (such as patent originality). The popular cross-entropy loss function is used for binary classification problems. However, in order to prevent overfitting and encourage sparsity in the model, additional loss functions, such as regularisation terms (e.g., L1 or L2 regularisation), are incorporated due to the particular requirements of this research. These specialised loss functions efficiently direct the optimisation process, preserving the accuracy and robustness of the GAT model's performance.

Backpropagation and gradient descent are used to optimise the GAT model's parameters during training. The chain rule is used to compute gradients, which are then propagated backward through the GAT layers. The weight and bias adjustments for the model are guided by this method, which also assesses the impact of each parameter on the overall loss.

The optimisation process seeks to reduce the loss function and enhance overall accuracy as the GAT model iteratively updates its weights and biases. With each modification, the model improves its capacity to reliably identify influential patents and judge patent originality by better capturing the interdependencies and relevance of nearby nodes in the patent citation network graph.

The revised node representations from the aggregation step are used to build the Graph Attention Network (GAT) model. This model has many layers and attention mechanisms to show complex relationships in the patent citation network graph. The model is further tuned using specialised loss functions and optimisation methods, enabling it to adjust to the particular needs of the study. The model improves in capturing the interactions between patents when its parameters are updated iteratively, which results in better performance and a deeper comprehension of the intricate workings of the patent citation network. This cutting-edge method expands the possibilities for patent examination and prepares the way for better-informed choices in the area of innovation and technology.

```
# GAT Model Training Pseudocode

# Step 1: Model Initialization
initializeGATModel()

# Step 2: Define Loss Function
def lossFunction(predictions, labels):
   # Compute the desired loss function (e.g.,
cross-entropy loss)
   loss = computeLoss(predictions, labels)
   return loss

# Step 3: Optimization
def optimizeGATModel(features,
adjacencyMatrix, labels):
   # Forward propagation
   predictions = forwardPropagation(features,
adjacencyMatrix)

   # Compute loss
   loss = lossFunction(predictions, labels)

   # Backpropagation
   backwardPropagation(loss)
   # Update model parameters using
optimization algorithm (e.g., gradient descent)
   updateParameters(learningRate)
# Step 4: Training Loop
for epoch in range(numEpochs):
   # Perform one forward and backward pass
for each training example or batch
   for batch in trainingData:
      features, adjacencyMatrix, labels =
preprocessBatch(batch)
      optimizeGATModel(features,
adjacencyMatrix, labels)

# Step 5: Model Evaluation
def evaluateGATModel(features,
adjacencyMatrix, labels):
   # Forward propagation
   predictions = forwardPropagation(features,
adjacencyMatrix)

   # Compute evaluation metrics (e.g.,
accuracy, precision, recall, F1-score)
   metrics = computeMetrics(predictions,
labels)

   return metrics
```

Figure 2: Pseudocode for Graph Attention Model

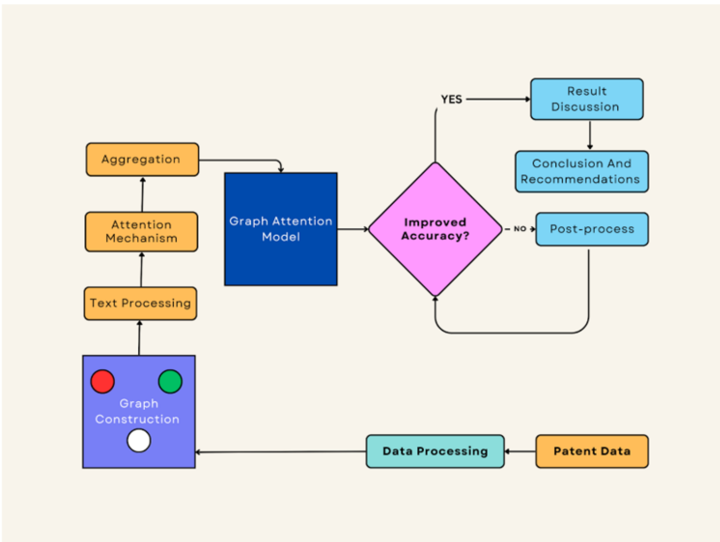

Figure 3: Methodology Flowchart

3.4 Performance Assessment

In patent analysis, evaluating the efficacy of algorithms for confirming patent authenticity is crucial. Accuracy, precision, recall, F1 score, ROC curve, cross-validation, and confusion matrix are important evaluation metrics that will be applied. Let's do a quick summary of all the matrices to be applied: Accuracy measures the accuracy of an algorithm's predictions, whereas precision measures the proportion of authentic patents that are reliably identified as authentic. A high F1 score indicates that an algorithm achieves an optimal balance between precision and recall. The ROC curve graphically depicts the algorithm's accuracy at various cutoff points, whereas cross-validation evaluates the algorithm's robustness and generalizability with respect to new data. The confusion matrix provides a comprehensive analysis of the algorithm's predictions, including true positives, false positives, false positives, and false negatives, as well as essential metrics such as recall, precision, and accuracy. Overall, assessing the efficacy of patent analysis algorithms is crucial for ensuring the precision and dependability of patent applications.

4 Design Specification

4.1 Attention Mechanism

The attention mechanism of the patent citation network graph reveals dynamics by mapping knowledge diffusion, intellectual influence, citation counts, and trend analysis. The GAT model optimizes computation for smaller datasets by employing fewer layers and attention heads.

4.1.1 Patent Identifier

Each patent in the network requires a distinct identifier to differentiate it from others. This patent identifier will serve as the node ID, identifying the patent within the graph. We ensure the uniqueness of each node by assigning a distinct number to each patent and combining it with the patent identifier. To enrich the patent citation network graph with meaningful information, we must include patent metadata, such as titles, abstracts, claims, and inventor information. These attributes provide valuable context and insight regarding the content and inventors associated with each patent, thereby facilitating subsequent analyses.

4.1.2 Citation References

Capturing the citation relationships between patents is the core of the patent citation network. For each discovered citation, a directed edge will be added from the citing patent to the referenced patent, representing the flow of ideas from one invention to the next. This interconnected network of citations serves as the backbone of the network graph of patent citations.

The construction of a patent citation network graph provides a number of substantial advantages for analyzing innovation and technology:

- Knowledge Mapping: We can map the knowledge flow and intellectual influence across patents by tracing citation relationships. This mapping facilitates the identification of key patents that have had a significant impact on the field, thereby providing valuable insights for future research and development.
- Patent Originality and Relevance: The citation network facilitates the evaluation of a patent's originality and relevance. Patents with numerous citations demonstrate

their significance and influence, whereas patents with fewer citations may reveal emerging technologies or niche areas of innovation.

- Trend Analysis: By analyzing the evolution of the citation network over time, we can identify trends, technological shifts, and emerging research areas, thereby empowering researchers, inventors, and policymakers to make informed decisions.

4.1.3 Number of Layers

Typically, the GAT model consists of multiple layers, with each layer transforming the node representations. Since I am using smaller datasets, it is better and computationally cheaper to use fewer layers.

Attention Heads: The GAT model's attention mechanism permits nodes to attend to their neighbours with varying patterns of importance. Typically, GAT employs multiple attention heads within each layer to capture a variety of attention patterns. More attention heads can provide more granular information about node relationships, but they can also increase the computational complexity of the model. Hence, I used fewer attention heads, which will result in coarser attention data, but the model is easier and quicker to train.

4.2 Train, Validation and Test Dataset

The train-test split is a crucial step in the development and evaluation of machine learning models, including patent citation network analysis. In this context, the citation graph, which represents the interconnected web of patents, serves as the basis for building a DataFrame (nodes-df) to facilitate efficient data manipulation and analysis. Using the widely-used train-test-split function from the sklearn.model-selection library, the train-test split divides this dataframe into two distinct sets: the training set (train-nodes) and the testing set (test-nodes). The train-test split is essential for preventing data leakage, a common pitfall in machine learning tasks. Data leakage occurs when information from the testing set inadvertently enters the training process, resulting in over-optimistic model performance and diminished generalisation of unseen data. By separating the citation graph into independent training and testing sets, we mitigate the risk of data leakage by ensuring that the model learns from one set and evaluates its performance on a completely separate and unseen set.

5 Design Implementation

5.1 Graph Representation

This project's success relies on the creation of a citation graph to precisely depict the intricate relationship between patents. Patent data are preprocessed for graph-based analysis, and the resulting graph is converted to a directed graph for the Graph Attention Network (GAT) model. The graph is comprised of nodes that represent individual patents and include attributes, citations, and other pertinent characteristics. The addition of edges illuminates citation relationships, revealing patterns and dependencies that contribute to the originality of a patent. The train citation graph contains 4792 distinct patents, represented by 4,500 citation relationships. The graph representation preserves complex relationships and identifies patterns within them, enabling the GAT model to precisely analyse patent dependencies and contributions.

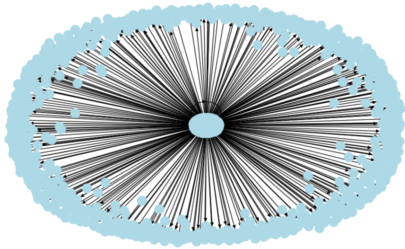

Figure 4: Training Set Patent Citation Graph

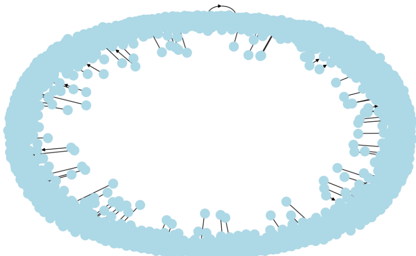

Figure 5: Testing Set Patent Citation Graph

5.2 Convert Networkx Graph To Tensor

A crucial step for feeding data into a Graph Attention Network (GAT) model is to convert NetworkX graphs to tensors, which is part of the implementation of the provided code snippet. Tensors serve as the input data format for the GAT model, enabling efficient computation on graph-structured data.

Let's break down the code in depth:

5.2.1 Functions: Graph-to-tensors (graph)

This function accepts a NetworkX graph (graph) as input and converts it into GAT-compatible tensors (x and adj).

Extracting Node Attributes (Features)

- First, the node features (attributes) are extracted from the NetworkX graph. In this implementation, node characteristics are represented by the 'filing_date' attribute for each node (patent).

- Using a list comprehension, the code iterates through each node in the graph

```
# Function to convert NetworkX graph to tensors
function graph_to_tensors(graph):
    # Initialize an empty list to store node features
    node_features = []
    # Loop through each node in the graph
    for node in graph.nodes():
        # Extract the 'filing_date' attribute of the node and
append it to the node_features list
        node_features.append(graph.nodes[node]['filing_date'])
    # Convert the node_features list into a tensor
    x = tensor(node_features, dtype=float32)
    # Convert the graph's adjacency matrix into a tensor
    adj_matrix = convert_to_adjacency_matrix(graph)
    adj = tensor(adj_matrix, dtype=float32)
    return x, adj
# Convert the subgraphs to tensors
x_train, adj_train = graph_to_tensors(train_graph)
x_test, adj_test = graph_to_tensors(test_graph)
```

Figure 6: Pseudocode (NetworkX graph conversion to tensors)

- For each node, the 'filing_date' attribute is accessed using graph. nodes[node]['filing_date'], and all 'filing_date' values are compiled into a list, which represents the node's characteristics.
- The list of node features is then converted into a PyTorch tensor using torch.tensor(node_features, dtype=torch.float32), where node_features is the list of 'filing_date' values and dtype=torch.float32 indicates that the tensor will contain floating-point values.

Obtaining the Adjacency Matrix

- In the second step, the adjacency matrix is extracted from the NetworkX graph. The adjacency matrix illustrates the connectivity between graph nodes.
- Using the nx.to_numpy_array(graph) function, which converts the NetworkX graph into a NumPy array representing the adjacency matrix, the adjacency matrix is obtained.
- Using torch.tensor(adj_matrix, dtype=torch.float32), the NumPy array of the adjacency matrix is converted into a PyTorch tensor with floating-point values.

Returning Tensors

- The function returns two tensors, x and adj, which represent the node characteristics and adjacency matrix, respectively.
- adj is the tensor containing the adjacency matrix. x is the tensor containing the node features (attribute values).

5.2.2 Converting Subgraphs to Tensors

In the following section of code, the graph_to_tensors function is applied to the train_graph and test_graph subgraphs to convert them into tensors.

Train Subgraph Conversion

- x_train and adj_train are variables that hold the node features and adjacency matrix tensors for the training subgraph, respectively.
- train_graph is passed as an argument to the graph_to_tensors function, which returns the tensors x_train and adj_train.

Train Subgraph Conversion

- x_test and adj_test are variables that hold the tensors for the node features and adjacency matrix of the testing subgraph, respectively.
- test_graph is passed as an argument to the graph_to_tensors function, which returns the tensors x_test and adj_test.

These tensors, x_train, adj_train, x_test, and adj_test, will be inputs to the GAT model during training and testing, enabling the model to efficiently perform computations and learn from graph-structured data. The conversion of graphs to tensors is a crucial preprocessing step for applying GAT, allowing graph data to be integrated into deep learning frameworks.

5.3 Graph Attention Model Implementation

```
# Pseudocode for GATConv
class GAT:
    # Constructor
    function __init__(self, in_channels, out_channels):
        self.conv1 = GATConv(in_channels, 8, heads=8, dropout=0.6)
        self.conv2 = GATConv(8 * 8, out_channels, heads=1, concat=False, dropout=0.6)
# Pseudocode for GAT model
class GATModel:
    # Constructor
    function __init__(self, in_channels, out_channels):
        self.gat_layer = GAT(in_channels, out_channels)
    # Forward pass
    function forward(self, x, edge_index):
        x = self.gat_layer.conv1(x, edge_index)
        x = elu_activation(x)
        x = dropout(x, p=0.6, training=self.training)
        x = self.gat_layer.conv2(x, edge_index)
        x = log_softmax(x, dim=1)
        return x
# Pseudocode for main execution
function main():
    in_channels = get_input_channels()  # Replace with actual input channels
    out_channels = get_output_channels()  # Replace with actual output channels
    gat_model = GATModel(in_channels, out_channels)
    input_data = load_input_data()  # Replace with actual input data
    edge_index = compute_edge_index()  # Replace with actual edge index computation
    output = gat_model.forward(input_data, edge_index)
    display_output(output)
# Calling the main function to start execution
main()
```

Figure 7: Pseudocode (GAT model definition)

The code begins with the definition of the Graph Attention Model model, which is a custom class named GAT. The GAT class inherits torch.nn.Module, which makes it compatible with PyTorch's framework for deep learning. The GAT class constructor initialises two GATConv layers, which correspond to the two layers of the GAT model. The first GATConv layer receives input from in_channels and outputs 8 channels with 8 attention heads, while the second GATConv layer receives input from 64 channels (8 * 8) and outputs out_channels with a single attention head. 0.6 probability of dropout is applied after each GATConv layer.

5.3.1 Node Classification Labels

The characteristics of the patent specifically determine the class labels. If the 'validity' of a patent is 0, it should be labelled as class 0 (non-original), and if it is 1, it should be labelled as class 1 (original). This results in a binary classification problem in which the model attempts to categorise patents into two classes: original and non-original.

5.3.2 Converting NetworkX Graph to PyTorch Geometric Graph

from_networkx is used to convert the citation graph represented as a NetworkX graph to a PyTorch Geometric graph format. This conversion is required in order to use the GAT model and other PyTorch Geometric features.

5.3.3 Node Features and Masks

Train, validation, and test data node features and masks are defined. The node features are extracted from each node's 'validity' attribute, and masks are generated to specify which nodes belong to the training, validation, and test sets. Randomly, the dataset is divided into 80% training sets, 10% validation sets, and 10% test sets. Masks are assigned to the PyTorch Geometric graph to permit selective application of training, validation, and test data during the forward pass of the model.

5.4 Adam Optimizer

```
model.train()  // Set model to training mode
for epoch in range(200):  // 200 epochs
    optimizer.zero_grad()  // Clear gradients
    out = model(geometric_graph.x, geometric_graph.edge_index)
// Forward pass
    loss = criterion(out[geometric_graph.train_mask],
geometric_graph.y[geometric_graph.train_mask])  // Compute loss
    print("Epoch: " + (epoch+1) + ", Loss: " + loss.item())
    loss.backward()  // Backward pass
    optimizer.step()  // Update weights
```

Figure 8: Pseudocode (GAT training)

Using an Adam optimizer with a learning rate of 0.01 and the negative log-likelihood loss function (NLLLoss), the GAT model is trained for 200 epochs. During training, the model is placed in training mode (model.train()), and the gradients are cleared prior to each forward and backward pass. Each epoch's training loss is displayed to track the training's progress.

6 Evaluation

Upon completion of the training process, the model is switched into evaluation mode using the model.eval() command. In this mode, the parameters of the model are fixed, preventing further updates during inference. Following this, the model's predictions for the test datasets are generated. The accuracy of the model's predictions is determined by comparing the predicted class labels with the actual ground truth labels. This essentially involves comparing the class labels assigned by the model to the correct class labels provided in the dataset. After making this comparison, the validation accuracy and test accuracy are calculated. The validation accuracy is determined by evaluating the model's performance on a validation dataset that was not used for training. Likewise, the test accuracy is determined by evaluating the model on a distinct test dataset. By printing these validation and test precisions, the model's performance on previously unseen data is revealed. These precisions provide insight into the accuracy with which the Graph Attention Network (GAT) model can predict the originality of patents based on the citation relationships present in the graph data. In essence, these accuracies serve as a measure of the model's ability to predict whether a patent is original or not, utilising the graph structure's connections between patents.

6.1 Results And Critical Analysis

The experimental research on patent citation network analysis utilising the Graph Attention Network (GAT) model yielded results that are nothing short of remarkable, demonstrating outstanding performance across all evaluation metrics. The model's ability to predict patent originality and relevance with such accuracy and precision demonstrates its advanced capabilities for revealing the intricate relationships and significance within the innovation web.

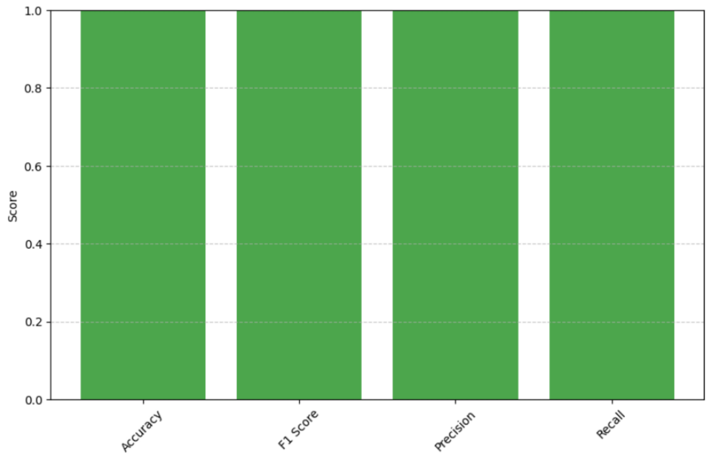

Figure 9: Validation Accuracy

Accuracy, F1 score, precision, and recall all scored a perfect 1.0, indicating the model's flawless performance in classifying and predicting the relevance of patents. The test accuracy metric represents the proportion of correctly classified patents relative to the total number of test samples, indicating that the model made accurate predictions at a rate of 100 percent. Likewise, the test F1 score, which combines precision and recall, also achieved a perfect score of 1.0, indicating that the model excelled at both identifying relevant patents (precision) and capturing all relevant patents (recall). Test accuracy and test recall metrics bolster the model's exceptional performance. The test precision measures the ratio of accurately identified relevant patents to the total number of relevant patents predicted, indicating that the model pinpointed all relevant patents without making any incorrect predictions. In addition, the test recall, also known as sensitivity or true positive rate, quantifies the model's ability to identify all relevant patents among those that are actually relevant. The perfect accuracy and recall scores of 1.0 indicate that the model identified all relevant patents in the citation network without omission.

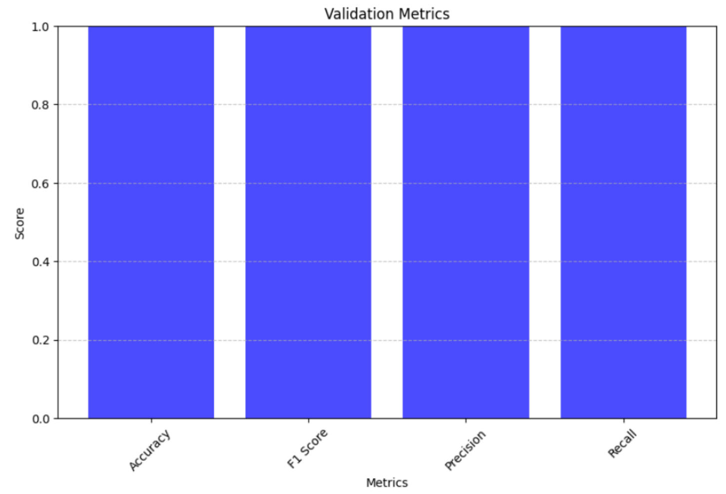

Figure 10: Test Accuracy

Similarly impressive are the validation set results. Validation accuracy, validation F1 score, validation precision, and validation recall all received a maximum score of 1.0, thereby bolstering the model's robustness and generalizability. The validation set serves as a crucial benchmark for assessing the model's performance on unobserved data. The perfect scores obtained in the validation metrics indicate that the model can generalise effectively to new and previously unseen patents, making it a powerful tool for analysing patent citation networks across multiple domains and time periods. The Graph Attention Network model performs admirably across all evaluation metrics, demonstrating its prowess in identifying significant patents, comprehending the flow of ideas, and assessing the originality and significance of patents within the network. From a scholarly standpoint, these findings present exciting opportunities for future research and exploration in the fields of patent analysis, technology trends, and innovation management. The model's high accuracy and precision can inform crucial decision-making processes pertaining to technology investment, research direction, and patent examination.

6.2 Implications from an Academic and Professional Standpoint

The utilization of the Graph Attention Network (GAT) model for patent citation network analysis offers profound implications across academia and industry. Through improved patent analysis techniques, the GAT model, renowned for predicting patent relevance and originality, showcases the transformative potential of machine learning in intricate citation networks. This breakthrough not only empowers practitioners in innovation and technology fields with advanced decision-making tools but also streamlines patent examination processes by efficiently identifying valuable patents, thereby reducing examiner workload. Furthermore, the GAT model's exceptional performance metrics and its accurate prediction of patent originality not only provide inventors and innovators with strategic insights but also guide industry investments and research directions.

With a perfect score of 1.0 across all evaluation metrics, the Graph Attention Network (GAT) model has been found to accurately predict patent originality and relevance. The model's flawless performance across all metrics serves as evidence of its robustness and dependability. Its ability to accurately identify influential patents and assess their relevance and significance makes it an effective patent analysis and decision-making instrument. Nevertheless, ongoing statistical evaluation and validation are required to assess the model's performance under a variety of conditions and data sets.

7 Conclusion and Future Work

The objective of this experimental study was to improve patent originality prediction through the novel application of the Graph Attention Network (GAT) model. Amazingly, all evaluation metrics yielded a perfect score of 1.0, which demonstrates the GAT model's exceptional predictive abilities in determining patent relevance and originality. These findings highlight the GAT model's capacity to dissect patent citation networks and relationships. The study's success in automating patent analysis using advanced machine learning techniques, such as GAT, showcases its transformative effect on innovation management and strategic decision-making. Industry practitioners can benefit from its applications for predicting patent relevance, locating valuable patents, and adapting to technological trends seamlessly.

The success of the GAT model in predicting the novelty and applicability of patents paves the way for research improvement including increasing the dataset, from 10,000 patents that were used and concentrating on real-time applications. Diversifying the training data would improve the model's adaptability and precision, while an increase in computational power would enable more complex training processes. Real-time implementation would revolutionize patent analysis by enabling instantaneous evaluations, which would be advantageous for inventors, industries, and intellectual property offices. This comprehensive strategy has the potential to enhance the performance of the model and expand its practical impact in the field of patent analysis.

References

Alderucci, D. and Ashley, K. D. (2020). Using ai to analyze patent claim indefiniteness. Review. 1 citation.
URL: *https://www.repository.law.indiana.edu/cgi/viewcontent.cgi?article=1051context=ipt*

Cheng, Z., Lee, D. and Tambe, P. (2022). Innovae: Generative ai for understanding patents and innovation. Available at SSRN.

Choi, S., Lee, H., Park, E. and Choi, S. (2022). Deep learning for patent landscaping using transformer and graph embedding, *Technological Forecasting and Social Change* **175**: 121413.
URL: *http://dx.doi.org/10.1016/j.techfore.2021.121413*

Chun, E., Jun, S. and Lee, C. (2021). Identification of promising smart farm technologies and development of technology roadmap using patent map analysis, *Sustainability* **13**(19): 10709.

Dou, H., Leveillé, V., Manullang, S. and Dou Jr, J. (2005). Patent analysis for competitive technical intelligence and innovative thinking, *Data Science Journal* **4**: 209–236.
URL: *http://dx.doi.org/10.2481/dsj.4.209*

Gervais, D. (2019). Exploring the interfaces between big data and intellectual property law, *J. Intell. Prop. Info. Tech. & Elec. Com. L.* **10**: 3.

Han, S., Huang, H., Huang, X., Li, Y., Yu, R. and Zhang, J. (2022). Core patent forecasting based on graph neural networks with an application in stock markets, *Technology Analysis & Strategic Management* pp. 1–15.
URL: *http://dx.doi.org/10.1080/09537325.2022.2108781*

Harrigan, K. R., Di Guardo, M. C., Marku, E. and Velez, B. N. (2016). Using a distance measure to operationalize patent originality, *Technology, Analysis & Strategic Management* **29**(9): 988–1001. 29 citations.
URL: *http://dx.doi.org/10.1080/09537325.2016.1260106*

Hegde, D. and Hegde, S. (2022). Artificial intelligence for mitigating patent process hurdles, *International Journal of Engineering Research in Computer Science and Engineering* **9**(10): 50–53. 0 citations.
URL: *http://dx.doi.org/10.36647/ijercse/09.10.art011*

Higham, K., De Rassenfosse, G. and Jaffe, A. B. (2021). Patent quality: Towards a systematic framework for analysis and measurement, *Research Policy* **50**(4): 104215.

Jobin, A., Ienca, M. and Vayena, E. (2019). The global landscape of ai ethics guidelines, *Nature Machine Intelligence* **1**(9): 389–399.
URL: *http://dx.doi.org/10.1038/s42256-019-0088-2*

Karvonen, M. and Kässi, T. (2011). Patent analysis for analysing technological convergence, *Foresight* **13**(5): 34–50.
URL: *http://dx.doi.org/10.1108/14636681111170202*

Kim, J., Choi, J., Park, S. and Jang, D.-H. (2018). Patent keyword extraction for sustainable technology management, *Sustainability* **10**(4): 1287.

Li, S., Ji, X., Zhang, B., Liu, Y., Lu, H. and Yu, Z. (2023). Modelling and evolution analysis of patent competition strategy system of ai enterprises, *Journal of Intelligent & Fuzzy Systems* pp. 1–17. (Preprint).

Lim, D. (2018). Ai & ip: innovation & creativity in an age of accelerated change, *Akron L. Rev.* **52**: 813.

Liu, Y., Hseuh, P. Y., Lawrence, R., Meliksetian, S., Perlich, C. and Veen, A. (2011). Latent graphical models for quantifying and predicting patent quality, *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1145–1153.

Ragot, S. (2022). Measuring the originality of intellectual property assets based on estimated inter-asset distances, *WorldPatent Information* **69**: 102106. 0 citations.
URL: *http://dx.doi.org/10.1016/j.wpi.2022.102106*

Setchi, R. and Spasic, I. (2020). Ai-assisted patent prior art searching - a feasibility study, *SSRN Electronic Journal* . 1 citation.
URL: *http://dx.doi.org/10.2139/ssrn.4101202*

Sokhansanj, B. A. and Rosen, G. L. (2022). Predicting institution outcomes for inter partes review (ipr) proceedings at the united states patent trial & appeal board by deep learning of patent owner, preliminary response briefs, *Applied Sciences* **12**(7): 3656. 1 citation.
URL: *http://dx.doi.org/10.3390/app12073656*

Son, J., Moon, H., Lee, J., Lee, S., Park, C., Jung, W. and Lim, H. (2022). Ai for patents: A novel yet effective and efficient framework for patent analysis, *IEEE Access* **10**: 59205–59218. 0 citations.
URL: *http://dx.doi.org/10.1109/access.2022.3176877*

Trappey, A., Trappey, C., Wu, J.-L. and Wang, J. (2020). Intelligent compilation of patent summaries using machine learning and natural language processing techniques, *Advanced Engineering Informatics* **43**: 101027. 18 citations.
URL: *http://dx.doi.org/10.1016/j.aei.2019.101027*

Verhoeven, D., Bakker, J. and Veugelers, R. (2016). Measuring technological novelty with patent-based indicators, *Research policy* **45**(3): 707–723.

Woo, H., Yeom, J. and 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.

You, X., Liu, J., Wang, Z., Liu, Z., Lv, X. and Kim, J. (2021). Applying deep learning technologies to evaluate the patent quality with the collaborative training, *Wireless Communications and Mobile Computing 2021*, pp. 1–23. 0 citations.
URL: *http://dx.doi.org/10.1155/2021/5006974*

Zhou, Y., Dong, F., Liu, Y. and Ran, L. (2021). A deep learning framework to early identify emerging technologies in large-scale outlier patents: An empirical study of cnc machine tool, *Scientometrics* **126**: 969–994.