ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Detecting Upcoming Patent Keywords by Predicting Keyword Trends Using Patent Keyword Network

Krisanapong Yodprechavigit[1] and Sukree Sinthupinyo[2]

ABSTRACT

In this ever-changing technological landscape, the ability to quickly predict technological trends becomes crucial for any company or institute engaged in informed decision-making and strategic planning. Data for predicting technological trends can come from various sources such as patent data, which is easily accessible to the public due to the nature of patents. This research is aimed at patent analysis, focusing on combining the keyword-based method, social network analysis (SNA) method, and neural network prediction to propose a feasible keyword trend prediction method based on patent analysis by targeting upcoming keyword trends. More specifically, we utilize Long Short-Term Memory (LSTM) to predict changes in keyword frequency using keyword centralities as input. To assess the effectiveness of the proposed method, we constructed the input dataset using the USPTO patent database in the Information and Communication Technology (ICT) field. We then experimented to compare the proposed method with the benchmark method. Furthermore, to counteract the unbalanced nature of patent data, the SMOGN method is introduced. The results demonstrate its potential for application in broader contexts.

Article information:

Keywords: Patent Analysis, Long Short-Term Memory, Social Network Analysis, Keyword Trend Prediction

Article history:
Received: December 8, 2023
Revised: February 1, 2024
Accepted: March 28, 2024
Published: April 20, 2024 (Online)

DOI: 10.37936/ecti-cit.2024182.255029

1. INTRODUCTION

A patent is a type of intellectual property that grants inventors the rights to their inventions in exchange for making publicizing their innovations. This public availability has led many researchers to explore their potential uses, resulting in the emergence of the research field known as patent analysis. It involves the process of extracting insights, trends, and information from patents within a specific field of technology or industry.

The initial approach to patent analysis primarily employed a network-based method [1], often utilizing citation networks [1-4]. Some researchers have also employed Social Network Analysis (SNA) techniques to analyzing these networks [4]. However, the network-based approach had limitations, as citation networks only captured the relationships between patents but not the information within them [1].

To address this limitation, researchers introduced a text-mining approach known as the keyword-based method. This method entails representing each patent using keywords that best describe its content, employing text-mining techniques [1]. These extracted keywords can subsequently undergo further analysis. Many keyword-based research often leads to predicting technological trends, indicating whether a particular technology is likely to gain or lose in popularity [6-8].

However, these predictions typically do not include the magnitude of growth or decline of the technology in question [6,7]. To overcome this limitation, one proposed solution involves using author-defined keyword features as input for Long Short-Term Memory (LSTM) [9] and regression models to predict future keyword frequencies [8]. This prediction allows for a more comprehensive assessment of the growth and decline of specific keywords.

This research aims to enhance keyword trend prediction methodologies by integrating features from patent keyword network analysis into Long Short-Term Memory (LSTM). Specifically, we aim to validate the superiority of this integrated approach over the method proposed by Lu et al.'s AKFP [9], which relies on author-defined keyword features. This study

[1,2] The authors are the Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Thailand, E-mail: k.yodprechavigit@gmail.com and Sukree.S@Chula.ac.th

aims to show that using features from patent network analysis improves keyword trend prediction for LSTM-based keyword trend prediction.

2. RELATED RESEARCH

2.1 Network-based Method

Based on Madani & Waber's research [1], patent analysis methodology has two approaches: network-based and keyword-based, as mentioned previously. However, it is essential to note that both approaches can be combined in the same study, as many keyword-based research studies also apply network-based analysis to examine keyword further. One illustrative example is the analysis of keyword co-occurrence networks.

Sternitzke *et al.* conducted network-based research with a primary focus on visualizing various types of patent networks within the field of Light-Emitting Diodes (LEDs) and Laser Diodes (LDs) [2]. The primary objective is identifying critical authors within cooperation and citation networks. This approach can be improved using social analysis methods, such as measuring centralities.

Cho & Shih's research focused on identifying core and emerging technologies in Taiwan [3]. The initial analysis consisted of descriptive statistical analysis, resulting in the identification of the top patent classes. Subsequently, the patent citation network was analyzed, resulting in the ranking of the most cited patent classes. Following this, a social network analysis method was employed to detect core technologies and emerging technologies. They used degree centrality to identify core technologies and examined structural holes to identify emerging technologies.

De Paulo & Porto analyzed technological routes and emerging technologies within patent cooperation networks [4]. The research used closeness and betweenness centrality to cluster citation networks into separate groups. Afterward, they analyze clusters individually to identify technological routes and uncover emerging technologies.

2.2 Keyword-based Method And Trend Analysis Using Machine Learning

Lee *et al.* conducted keyword-based research on ICT healthcare patents [5]. They used the bag-of-words model and TF-IDF to extract keywords from patents. Then, they constructed a network based on keyword co-appearance. Next, they divided the keyword network into clusters. These clusters were then analyzed to identify promising technologies in ICT-healthcare convergence services.

Kumari *et al.* focused on analyzing research topics in the field of humanoid robot technology [6]. The research involved applying both keyword co-occurrence analysis and social network analysis. Centrality analysis identified keywords with high centrality. Subsequently, they constructed the network at 3-year intervals to investigate the growth and decline of each research topic within the hype cycle.

Balili *et al.* aim to track and predict the evolution of research topics in research publications [7]. They extracted keywords from patents and constructed a dynamic keyword co-occurrence network. The first part of the research involved tracking the evolution of topics by categorizing communities as 'survive,' 'growth,' 'shrink,' 'split,' or 'merge.' In the second part, they used structural and temporal features from the created communities to classify whether the community would persist or dissolve.

Lu *et al.* used Author-Defined Keyword Frequency Prediction (AKFP) to detect trends in research topics [8]. The prediction model employed a Long Short-Term Memory (LSTM) with temporal features, persistence, community size, and community development potential as inputs for the LSTM. The result is predictions for keyword frequencies in the future, thereby able to predict the magnitude of keywords' growth or decline.

2.3 Research Objective

The primary objective of this study is to predict the patent keyword trends along with their magnitude of changes by forecasting patent keyword frequency, which the current method of patent keyword trend prediction cannot predict well. The second goal is to explore the potential of utilizing an author-to-keyword relationship network called the author keyword co-occurrence network [10] in the patent analysis field, which will be called the patent author-keyword co-occurrence network. The last goal is to combine the concepts of keyword-based predictive trend analysis and keyword-based patent analysis while incorporating a qualitative method to predict the keyword trends and their magnitude of changes. On the prospect of patent analysis in this research, the keyword-based method is patent keyword extraction. On the other hand, constructing the patent author-keyword co-occurrence network from the extracted patent keyword is the network-based method.

To achieve these goals, we introduce five categories of metrics for each keyword node extracted from the patent author-keyword co-occurrence network. These features include Betweenness Centrality [11], Eigenvector Centrality [12], Closeness Centrality [11], Load Centrality [13], and PageRank [14]. We selected these features as inputs for LSTM [9] to predict the keyword trend. The objective is to demonstrate the impact of author keyword co-occurrence data on future keyword frequency and validate the feasibility of predicting the keyword trend in patent analysis.

3. PROPOSED METHODOLOGY

This paper presents the method for predicting patent keyword frequency using an LSTM [9] combined with SNA centrality [11-14] applied to a patent author-keyword co-occurrence network. We illustrated the steps for the proposed keyword frequency prediction method in Figure 1.

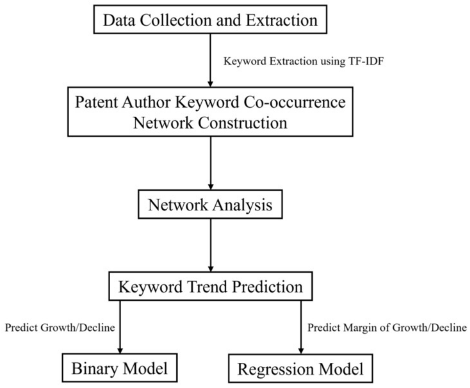

Fig.1: *Overall Method of Proposed Method.*

3.1 Data Collection And Extraction

The patent data was downloaded from the USPTO patent database, bulkdata.uspto.gov, in bulk XML files, covering the years from 2011 to 2021 [15]. Afterward, we extracted bibliography data and abstract text from each patent. The extracted bibliography data includes patent codes, corresponding technological codes known as Cooperative Patent Classification codes (CPC), and the author's name. Patents under CPC code G03C, on the information and communication field [16], were selected as the target group for further analysis. To extract the corresponding keywords from each patent, we used the text mining method TF-IDF [17] in the keyword extraction process. TF-IDF method is considered one of the popular data mining methods for patent keyword extraction [1]. Its performance is shown in [18].

We selected five keywords to be the corresponding keywords for each patent. The number five comes from our initial observation of patent keyword extraction using TF-IDF on our data. The most important keywords for each patent are commonly within the first five extracted keywords. Usually, the sixth and those following are typically common or gibberish words with no significant meaning.

3.2 Patent author-keyword co-occurrence network Construction

The relationship between authors and keywords is the basis of the patent author-keyword co-occurrence network. This relationship was established from the previously extracted patent data, linking the authors of the patents to the corresponding keywords of the patents. The result is in a list of pairs of author keywords for each year. Table 1 shows an example list from the year 2011.

Table 1: *Example of author-keyword pairs list for the year 2011.*

Author	Keyword
Miyoshi Kenichi	value
Miyoshi Kenichi	cir
Miyoshi Kenichi	rate
Miyoshi Kenichi	request
Miyoshi Kenichi	section
...	...
Zimmerman Jeffrey A.	ac
Zimmerman Jeffrey A.	powered
Zimmerman Jeffrey A.	element
Zimmerman Jeffrey A.	device
Zimmerman Jeffrey A.	data

We created networks using patent data from each year between 2011 and 2019, using them as input for the keyword trend prediction process. Figure 2 displays a sample network, illustrating the interconnections between each patent and keyword. The network represents Patent A as an author and its five corresponding keywords. Then, another patent is further added, with some may contain some of the exact corresponding keywords as patent A. These same keywords will then connect patents into a patent author-keyword co-occurrence network.

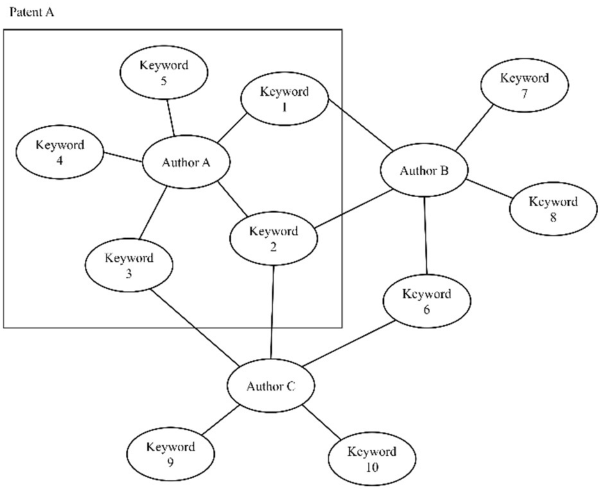

Fig.2: *Sample patent author-keyword co-occurrence network.*

3.3 Network Analysis

After constructing the network, we performed network analysis to gather information for further use as input data in the keyword frequency prediction process. We then used the Social Network Analysis (SNA) method to obtain metrics for the keyword nodes in the network we created. The obtained metrics included Eigenvector centrality, Closeness centrality, Betweenness centrality, Load centrality, and PageRank [11-14]. We selected Betweenness centrality, Eigenvector centrality, and Closeness centrality because they are frequently analyzed centrality in the SNA field [19]. We chose Load centrality as it is considered an improved version of Betweenness centrality [13]. We opted for PageRank because it effectively represents the importance of the keyword node [20].

3.3.1 Eigenvector Centrality

Eigenvector centrality is a metric used in social network analysis to quantify the importance and influence of network nodes. Eigenvector centrality is determined not only by their direct connections but also by the quality of their connections. Essentially, Eigenvector centrality evaluates how well a node connects to other well-connected nodes [12]. Assuming a network represented by an adjacency matrix A, eigenvector centrality is calculated by Equation 1.

$$Ax = \lambda x \tag{1}$$

Where A is the adjacency matrix of the network. x is the eigenvector centrality vector representing the centrality scores for all nodes in the network. λ(lambda) is the eigenvalue associated with this eigenvector.

3.3.2 Closeness Centrality

Closeness centrality is a network centrality measure that quantifies how quickly a node can reach all other nodes in a network. The core concept is that the shorter the average distance to other nodes, the more central it is in the network. A node with a high closeness centrality is considered central in the network [11]. Closeness centrality can be calculated by Equation 2.

$$C_i = \frac{(N-1)}{\sum_j d_{ji}} \tag{2}$$

Where C_i is the closeness centrality of node i. N is the total number of nodes in the network. d_{ji} is the shortest distance from node j to node i in the network.

3.3.3 Betweenness Centrality

Betweenness centrality is a network centrality measure that quantifies how well a node serves as a bridge within the network. A node with high betweenness centrality is critical to the network as it provides efficient pathways connecting different parts. Removing such nodes would result in longer paths or even disconnections between various network segments [11]. Betweenness centrality can be calculated by Equation 3.

$$g_i = \sum_{s \neq i \neq t} \frac{\sigma_{st}(i)}{\sigma_{st}} \tag{3}$$

Where g_i is the betweenness centrality of node i. σ_{st} represents the total number of shortest paths from node s to node t in the network. $\sigma_{st}(i)$ represents the number of those shortest paths that pass through node i.

3.3.4 Load Centrality

Load centrality, or 'Newman's Betweenness centrality,' is a variant of betweenness centrality proposed by Newman [13]. The critical difference between Betweenness centrality and Load centrality is the inclusion of all paths using a random walk method rather than using only the shortest path when calculating Betweenness centrality. Allowing Load centrality to identify nodes with high centrality that do not lie on the shortest path, which would be undetectable with Betweenness centrality [13].

Assuming a network represented by a diagonal matrix of vertex degrees (D) and an adjacency matrix (A), Load centrality can be calculated by the following process. First, construct the matrix (D-A) and remove a single row and its corresponding column. Then, invert the resulting matrix and add back the previously removed row and column, along with a new row and column consisting of all zeros. The resulting matrix called matrix T, with elements (T_{ij}). Load centrality can then be calculated by the following Equation 4-7.

$$b_i = \frac{\sum_{s<t} I_i^{(st)}}{\frac{1}{2}n(n-1)} \tag{4}$$

$$I_i^{(st)} = \frac{1}{2}\sum_j A_{ij}|T_{is} - T_{it} - T_{js} + T_{jt}|, \quad \text{for i} \neq \text{s,t}. \tag{5}$$

$$I_s^{(st)} = 1 \tag{6}$$

$$I_t^{(st)} = 1 \tag{7}$$

Where b_i is the load centrality of node I and n is the total number of nodes in a network.

3.3.5 PageRank

PageRank is an algorithm employed by search engines, such as Google, to rank web pages in search results. PageRank analyzes the link structure within a network and gauges the importance of each web page. Its core principle is to assess both the quantity and quality of the links leading to the current webpage, determining how significant the current webpage is [20]. In this sense, PageRank bears similarities to eigenvector centrality [14]. PageRank is computed by an iterative algorithm as Equation 8.

$$PR(A) = \frac{1-d}{N} + d\sum_{B \in M(A)} \frac{PR(B)}{L(B)} \tag{8}$$

Where PR(A) is the PageRank score of page A. d is a damping factor, typically set to around 0.85. It represents the probability that a user will continue randomly surfing rather than following links. N is the total number of pages in the network. $M(A)$ denotes all pages B that link to page A. $PR(B)$ is the PageRank score of page B, and $L(B)$ is the number of outbound links on page B. The formula will iteratively calculate the PageRank of all nodes in a network until they converge to a stable result.

3.4 Keyword Trend Prediction

The primary process for predicting keyword trends involves utilizing Long-Short Term Memory (LSTM) [9] to forecast the magnitude of change in keyword frequency and predict upcoming keywords using keyword centrality. Moreover, to address the unbalanced nature of the dataset, the SMOGN method [21] was employed to enhance the model's accuracy further.

3.4.1 Long Short-Term Memory (LSTM)

LSTM is a type of recurrent neural network (RNN) well-suited for handling time series data, such as patent data, in this research. An improvement of LSTM over standard RNN is its ability to mitigate the vanishing gradient problem, which arises when training networks over long periods. An LSTM network has three gates, a memory cell (C_t), and a hidden state (h_t). These three gates include an input gate (i_t), a forget gate (f_t), and an output gate (o_t) [9]. Each component is explained in the mathematical Equations 9-13.

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \quad (9)$$

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \quad (10)$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \quad (11)$$

$$C_t = f_t C_{t-1} + i_t \tanh(w_c[h_{t-1}, x_t] + b_c) \quad (12)$$

$$h_t = o_t \tanh C_t \quad (13)$$

Where σ is the sigmoid activation function. w represents the weight matrix for each component. h_{t-1} is the previous cell state, and x_t is the current input. For the loss functions used in this research, we applied Mean Squared Error (MSE) [22] in regression model and employed Binary Cross-entropy (BC) [23] in binary classification model. Both loss functions can be calculated by Equations 14 and 15.

$$MSE = \frac{1}{n}\sum\nolimits_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (14)$$

$$BC = -\frac{1}{n}\sum\nolimits_{i=1}^{n} y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i) \quad (15)$$

Where n is the number of data points, $\hat{y}_i$ represents the predicted value and y_i represents the actual target value.

3.4.2 Synthetic Minority Over-Sampling Technique for Regression with Gaussian Noise (SMOGN)

The SMOGN method is a data preprocessing technique used to address the issue of imbalanced datasets in regression problems. SMOGN combines random undersampling with two oversampling methods: the Synthetic Minority Over-Sampling Technique for Regression (SMOTER) and Gaussian Noise. SMOTER is adapted from the SMOTE method to address regression problems [24] specifically.

In essence, SMOGN works by undersampling the majority data and oversampling the minority data through augmented data. To generate augmented data, each minority data point is selected along with its k neighbors. The neighbors are interpolated within a defined safe range using the SMOTER method. For neighbors located outside this safe range, the SMOTER method is deemed unsuitable. Instead, new examples are generated by applying Gaussian Noise to the selected data. The safe range is defined as half the median distance between the data point and its k neighbors [21]. In this study, we utilized the Python library 'smogn' [27].

4. EXPERIMENTAL SETUP

4.1 Data

We collected patent grant bibliographies and abstract data in XML format from 2011 to 2021 through bulk downloads from bulkdata.uspto.gov [15]. Specifically, we selected patents related to information and communication technology using CPC code G03C [16], resulting 8,140 patents. Next, we extracted author information and abstracts for each patent, totaling 14,621 authors. Additionally, we assigned five keywords to each patent using the TF-IDF method [17] applied to their abstracts using the Scikit-learn library [25], resulting 7,298 keywords.

4.2 Patent author-keyword co-occurrence network

We used the Python library NetworkX [26] to generate the patent author-keyword co-occurrence network. This library took the relationship tables as input to create the network for each year. Subsequently, the keyword centrality data for each year was calculated by applying social network analysis methods across the years with the NetworkX library [26]. The calculated keyword centrality data includes the following metrics: Eigenvector centrality, Closeness centrality, Betweenness centrality, Load centrality, and PageRank [11-14].

4.3 Setup Training Set

The most effective model for predicting keyword frequency involves short-term predictions, which forecast keyword frequency for two years in the future based on input data [8]. This model is combined with a moving window method with a three-year length to create the training set for the LSTM [9]. However, the target keyword frequency was imbalanced, with most results having keyword frequencies close to zero. In total, there are 1,466 keyword rows in the positive group, 2,281 keyword rows in the negative group, and 21,330 keywords in the unchanged group. The SMOGN method [21] was applied to the positive and negative groups to address this imbalance.

Finally, the data in each group was divided into training sets, verification sets, and test sets in an 8:1:1 ratio, respectively. The input dataset for the binary classification model was then created by classifying the keyword slope into the TRUE (with a slope >1, signifying them as upcoming keywords) and FALSE (with a slope <=1, signifying them as unchanged-to-decline keywords). Figure 3 shows the overview of the training set.

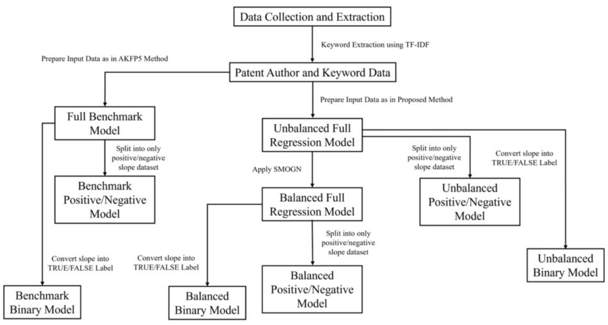

Fig.3: *Overview of training set setup.*

We will provide further explanations of these steps below.

4.3.1 Moving Window

To optimize time series data, particularly patent keyword data, we employed the moving window method with a fixed step [8], selecting a window size of three steps. The input dataset consists of the proposed centrality values for three consecutive years and the corresponding keyword slope. For example, starting with the first row for keyword A, the first 15 columns are filled with five centralities for years 1, 2, and 3 consecutively. We filled the last column with the target slope, which we calculated by subtracting the frequency of keyword A in year three from its frequency in year five. We then repeated this process for every keyword.

After that, the starting year is moved up by 1. To fill the first 15 columns of the next row, we used five centralities for years 2, 3, and 4 of the keyword A. We then calculate the target slope by determining the difference in frequency of keyword A between years four and six. The process continued until no more data was available to construct the dataset for the following year. In our case, the input dataset starts from the year 2011 and goes up to 2017, as the available data only spans from 2011 to 2021.

Figure 4 shows an overview of the moving window operation. The centrality table results from centralities extracted from the patent author-keyword co-occurrence network. It contains five centralities of every keyword each year. For example, A2011 contains five centralities of keyword A in 2011. Input for LSTM is then created by layering C_1, C_2, and C_3 on input table shown in Figure 5.

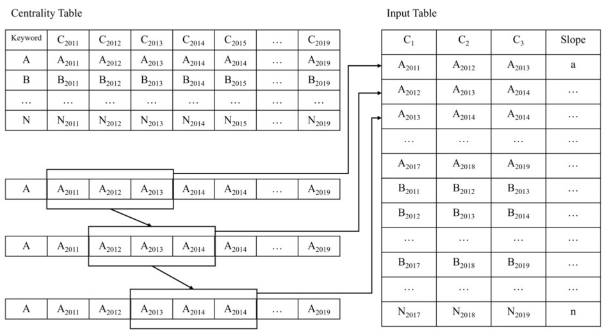

Fig.4: *Overview of how the moving window operates.*

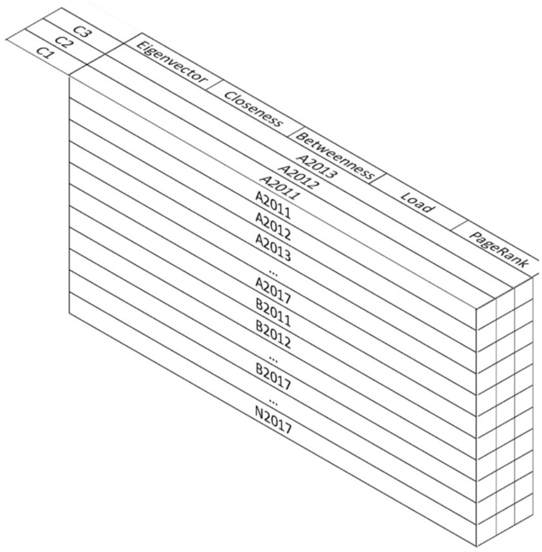

Fig.5: *LSTM input data.*

4.3.2 Keyword Category

First, input rows with no centrality data, meaning that the specific keyword does not appear even once within a moving window of 3 years, were removed from the input dataset. To tackle the imbalance issue mentioned earlier, we categorized each keyword for every year into one of three groups: positive, negative, or unchanged. We determined the grouping by comparing each keyword's frequency in the current year to its frequency from two years prior. We classify keywords with an increase in frequency of more than one as positive, keywords with a decrease in frequency of more than one as negative, and keywords with a change of less than one as unchanged.

Subsequently, the target for LSTM prediction shifted from keyword frequency to the change in keyword frequency, referred to as the 'keyword slope' from now on. The LSTM prediction model was then applied to the entire dataset and separately, with focusing on the results for the positive group. This is because keywords with an increased frequency are more likely to represent upcoming trends, especially those with higher slopes.

4.3.3 Keyword Balancing

Due to the nature of the patent data, the keyword slope skewed toward zero. Even after splitting the keyword categories, the keyword slope remains skewed toward 2 for the positive group and toward -2 for the negative. We then applied the SMOGN method [22] to both groups to further balance the dataset. After applying SMOGN, there were 5,717 keyword rows in the positive group, 3,647 keyword rows in the negative group, and 5,464 keywords in the unchanged group. This resulted in a more balanced dataset for both groups. Figures 6-8 display histograms of each dataset type before applying SMOGN, while Figures 9-11 show them after applying SMOGN. By comparing the histograms before and after applying SMOGN, it was evident that the large number of values near zero decreased while values farther from zero increased, thereby further balancing the dataset.

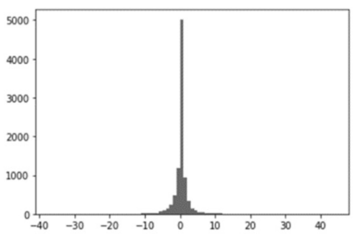

Fig.6: *Histogram of Full Dataset Before Applying SMOGN.*

5. EVALUATION

For the regression model, the four most popular criteria to evaluate the regression prediction model are as follows: Mean Squared Error (MSE) [22], Root Mean Squared Error (RMSE) [28], mean absolute error (MAE) [28], and coefficient of determination (R^2)

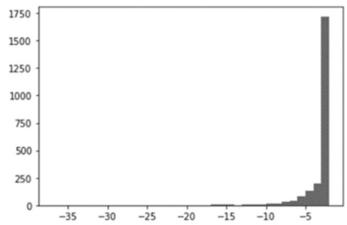

Fig.7: *Histogram of Negative Dataset Before Applying SMOGN.*

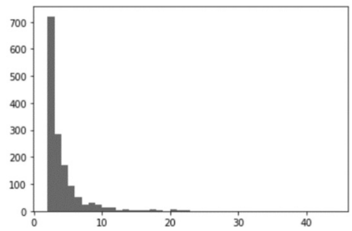

Fig.8: *Histogram of Positive Dataset Before Applying SMOGN.*

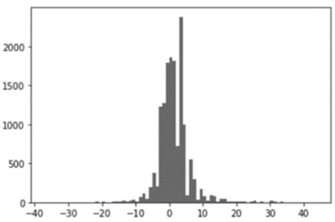

Fig.9: *Histogram of Full Dataset After Applying SMOGN.*

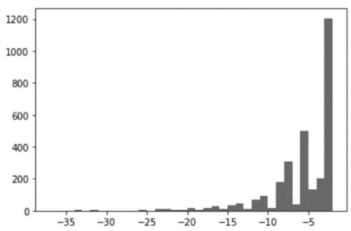

Fig.10: *Histogram of Negative Dataset After Applying SMOGN.*

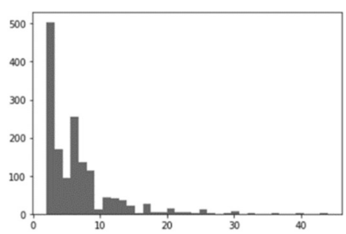

Fig.11: *Histogram of Positive Dataset After Applying SMOGN.*

[29]. MAE represents the average absolute error between the actual and predicted values. MSE is similar to MAE but is more sensitive to the variance between the actual and predicted values. RMSE is the square root of MSE. In an ideal case, RMSE should be equal to MAE. However, RMSE inherits the sensitivity to larger errors from MSE. Therefore, the more significant the gap between MAE and RMSE, the larger the prediction error [28]. R^2 represents the correlation between the actual and predicted values, ranging from 0 to 1 [29]. The ideal outcomes include low MAE and MSE values with high R^2. MAE, RMSE, and R^2 are defined by following equation 16-18.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{16}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{17}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{18}$$

Where n is the number of data points. $\hat{y}_i$ represents the predicted value. y_i represents the actual target value. $\bar{y}$ represents the average target value.

For the binary model, the criteria for model evaluation are the combination of accuracy [30], recall, and F1 score [31]. To assess the performance of the proposed method, we will compare it with the AKFP 5 model [8] based on the criteria we mentioned earlier. The reasons for selecting the AKFP 5 model are as follows: First, the AKFP 5 model is a prediction model that can be replicated by the patent data used in the proposed model, making it possible to compare the performance of both models using the criteria mentioned above. Second, the performance of the AKFP 5 model surpasses that of regular machine learning approaches, such as LR, KNN, XGBoost, and RF. Third, AKFP 5 performed best among the three proposed methods by Lu *et al.* [8].

6. PARAMETERS

In this research, we aim to keep the LSTM parameters as identical to the AKFP 5 method [8] as possible, with the primary objective of comparing the performance of our proposed method against the AKFP 5 method. From now on, we will refer to the AKFP 5 method as the benchmark method. By retaining similar LSTM parameters, we can assess the effectiveness of the proposed method compared to the benchmark method in an equitable manner. This way, we can attribute any observable differences in performance to the different input data rather than the model configuration. On the other hand, learning rates, batch size, and epochs still need to be tuned to maintain optimal performance on the benchmark method. We selected the Adadelta optimization algorithm [32] for the optimizer. Not only is the Adadelta method superior to the traditional gradient descent method, which has a problem with falling into local optima, but it also has an automatic learning rate adjustment algorithm that allows different parameters to have different learning rates. Table 2 provides a detailed description of the LSTM parameters used in the models for this research. For the framework used, we used the TensorFlow framework through the Keras library [33].

Table 2: *Parameter of LSTM Model.*

Parameter	Benchmark model	Proposed model (regression)	Proposed model (binary)
Number of units on each layer	256,512 (LSTM), and 1	256,512 (LSTM), and 1	256,512 (LSTM), and 1
Activation Function	ReLU	ReLU	ReLU
Learning Rates	0.1	0.1	0.1
Optimizer	Adadelta	Adadelta	Adadelta
Batch size	32	32	32
Epochs	200	200	200
Activation Function	-	-	Sigmoid

7. RESULTS AND COMPARISION

We divided the input dataset into three groups for the sake of comparison. These three groups are as follows: benchmark, unbalanced, and balanced. Figures 12 and 13 illustrate the differences in input datasets between the groups. In both the unbalanced and balanced models, we applied the proposed method. However, we used the SMOGN method only for the balanced model, not the unbalanced one. First, we compared the results of all three full models—the benchmark full model, the unbalanced full model, and the balanced full model. Table 3 presents the results, and Figure 12 shows a scatterplot. Next, we then compared the results of both the three positive and negative models. These included the positive/negative benchmark models, positive/negative unbalanced models, and positive/negative balanced models. Table 3 presents the results, and Figure 13 shows a scatterplot. Finally, we compared the results of three binary models: the binary benchmark, balanced, and unbalanced model. The results are presented in Table 10 with their respective confusion matrices, while the normalized confusion matrices are shown in Tables 4-9.

8. DISCUSSION

Based on the results, the balanced model achieved the lowest MSE and MAE in the positive group, making it the most effective method for the primary focus of this research. Moreover, the results of the

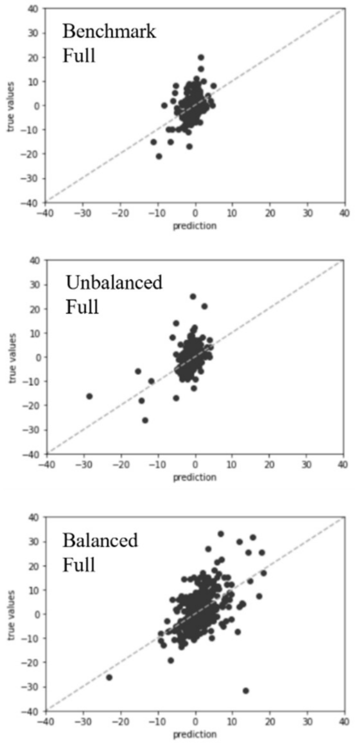

Fig.12: *Scatter Plot of Each Full Model.*

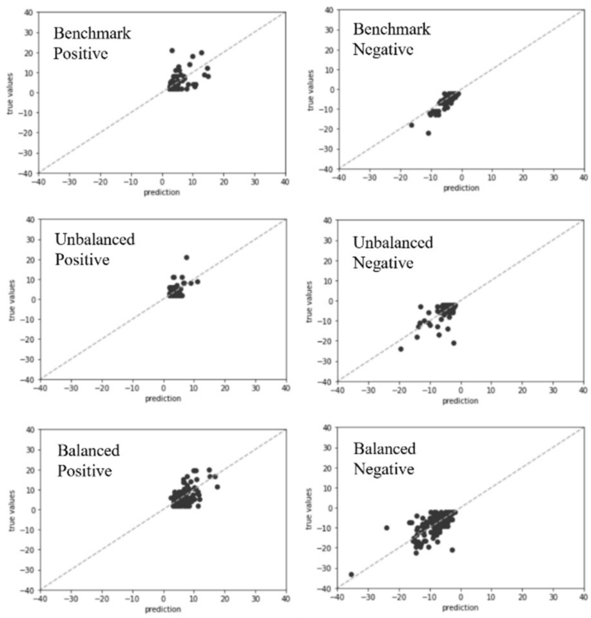

Fig.13: *Scatter Plot of Each Positive/Negative Model.*

Table 3: *Performance of each Regression Model.*

Result		Bench mark	Unbalanced	Balanced
Full	MSE	3.516	**2.776**	15.128
	RMSE	1.876	**1.667**	3.889
	MAE	1.134	**0.792**	2.607
	R^2	0.188	0.19	**0.257**
Positive	MSE	17.584	11.232	**10.362**
	RMSE	4.193	3.351	**3.219**
	MAE	2.167	**1.479**	2.538
	R^2	0.291	0.335	**0.364**
Negative	MSE	**2.427**	4.383	9.283
	RMSE	**1.558**	2.094	3.047
	MAE	**0.997**	1.006	2.155
	R^2	**0.709**	0.515	0.542

Table 4: *Confusion Matrix on Benchmark Binary Model.*

		Predict label	
		FALSE	TRUE
Actual label	FALSE	1444	0
	TRUE	122	0

Table 5: *Normalized Confusion Matrix on Benchmark Binary Model.*

		Predict label	
		FALSE	TRUE
Actual label	FALSE	0.922	0
	TRUE	0.078	0

Table 6: *Confusion Matrix on Unbalanced Binary Model.*

		Predict label	
		FALSE	TRUE
Actual label	FALSE	2351	0
	TRUE	157	0

Table 7: *Normalized Confusion Matrix on Unbalanced Binary Model.*

		Predict label	
		FALSE	TRUE
Actual label	FALSE	0.937	0
	TRUE	0.063	0

Table 8: *Confusion Matrix on Balanced Binary Model.*

		Predict label	
		FALSE	TRUE
Actual label	FALSE	802	100
	TRUE	331	250

Table 9: *Normalized Confusion Matrix on Balanced Binary Model.*

		Predict label	
		FALSE	TRUE
Actual label	FALSE	0.541	0.067
	TRUE	0.223	0.167

Table 10: *Performance of each Binary Model.*

Result	Benchmark	Unbalanced	Balanced
Accuracy	0.922	0.937	0.709
Recall	0	0	0.537
F1 Score	0	0	0.43

balanced dataset in both the full and negative categories were worse than those of the benchmark and unbalanced datasets. On the other hand, the unbalanced model had the lowest MSE and MAE in the full group. However, the difference is very little when compared to the benchmark model, as the difference in both MSE and MAE was lower than 1. Upon comparing the scatterplots of these results, we hypothesize that this may be due to the effect of the imbalanced dataset, which causes data points to cluster around a natural slope of zero. This characteristic becomes evident when we consider the results of the binary model. The F1 scores for both the benchmark and unbalanced datasets were at 0, while the balanced dataset demonstrates a notable improvement with an F1 score of approximately 0.5. When we consulted the F1 score and recall in conjunction with the confusion matrix, it was safe to assume that the leading cause of the low F1 score in the benchmark and unbalanced datasets stems from the skewness of both datasets toward the unchanged-to-decline keyword. This skewness in both the benchmark and unbalanced datasets was pivotal in achieving lower MSE and MAE than the balanced dataset in the full group. Furthermore, comparing the recall between the benchmark and balanced method shows two things. First, the balanced method performs better at predicting positive instances than the benchmark method. Second, the benchmark method fails to predict any positive instances at all. The benchmark method's lack of prediction ability may stem from the previously mentioned skewness. To mitigate the influence of data clustering in the positive and negative groups, we employed categorizing the dataset into three distinct keyword categories. By applying the LSTM model separately to the positive and negative categories, we aim to reduce the impact of data clustering, particularly from the unchanged group, on the predictive performance of both the positive and negative groups. This resulted in lower MSE and MAE in the positive category but higher MSE and MAE in the negative category when comparing the balanced dataset to the benchmark and unbalanced dataset.

9. CONCLUSIONS

In this study, we proposed a novel approach to predict the patent keyword trends along with the magnitude of changes. We proposed a method that combines a patent author-keyword co-occurrence network with a Long Short-Term Memory (LSTM). The LSTM used input features extracted from the network's keyword-related metrics and predicted the keyword slopes, which indicate future trends. Our finding is as follows: we can successfully predict the upcoming keywords with acceptable accuracy using the proposed method. While the performance of the proposed method did not significantly surpass existing benchmarks, it demonstrates competitive potential. Future research can further optimize and refine this potential. Moreover, we believe this study could shed light on the potential of applying co-occurrence networks and social network analysis (SNA) to predict keyword trends.

10. LIMITATION AND FUTURE STUDY

While this research proposes a new approach to patent keyword trend prediction, it is important to acknowledge several limitations and possibilities for improvement in future studies. First, our dataset is limited to the abstracts of patents categorized by the G03C CPC patent code. Future research could expand the scope using a larger dataset, including the full patent text rather than just abstracts or grouping patents based on research keywords instead of their assigned codes.

Text mining techniques can be improved by exploring alternative methodologies, implementing more robust data cleaning, and adopting effective keyword grouping strategies, following established guidelines [34]. Additionally, the constraint of using five keywords to represent each patent may not be optimal; varying the number of keywords for different patents could lead to more precise representations of each patent.

A notable limitation of our study is the lack of comparison between traditional keyword co-occurrence networks [6-7] and the author-keyword co-occurrence network proposed in this study. Such a comparison could shed light on which method is a more effective for patent analysis. Moreover, there are alternative time-series and classification models that could be used for evaluation, beyond the LSTM model employed in this study.

Furthermore, the input dataset can be expanded by including more centrality data or network metrics to improve the analysis. Additionally, future studies can delve into feature analysis to determine the significance of each centrality measure by employing feature analysis techniques such as leave-one-out modeling [8].

These limitations and future research directions provide opportunities to enhance the depth and breadth of patent analysis research, further contributing to the importance of the patent analysis field in the current data-driven business and society.

AUTHOR CONTRIBUTIONS

Conceptualization, K.Y. and S.S.; methodology, K.Y.; programming, K.Y.; validation, K.Y., S.S.; data curation, K.Y.; data analysis, K.Y.; writing-original draft preparation, K.Y.; writing-review and editing, K.Y., S.S.; visualization, K.Y.; supervision, S.S. All authors have read and agreed to published version of the manuscript.

References

[1] F. Madani and C. Weber, "The evolution of patent mining: Applying bibliometrics analysis and keyword network analysis," *World Patent Information*, vol. 46, pp. 32-48, September 2016.

[2] C. Sternitzke, A. Bartkowski and R. Schramm, "Visualizing patent statistics by means of social network analysis tools," *World Patent Information*, vol. 30, no. 2, pp. 115-131, June 2008.

[3] T.-S. Cho and H.-Y.Shih, "Patent citation network analysis of core and emerging technologies in Taiwan: 1997–2008," *Scientometrics*, vol. 89, pp. 795-811, July 2011.

[4] A. F. de Paulo and G. S. Porto, "Evolution of collaborative networks of solar energy applied technologies," *Journal of Cleaner Production*, vol. 204, pp. 310-320, December 2018.

[5] S. Lee, J. Choi and Y. W. Sawng, "Foresight of promising technologies for healthcare-IoT convergence service by patent analysis," *NIScPR Online Periodicals Repository*, vol. 78, no. 8, pp. 489-494, August 2019.

[6] R. Kumari, J. Y. Jeong, B. H. Lee, K. N. Choi and K. Choi, "Topic modelling and social network analysis of publications and patents in humanoid robot technology," *Journal of Information Science*, vol. 47, no. 5, pp. 658-676, December 2019.

[7] C. Balili, A. Segev and U. Lee, "Tracking and predicting the evolution of research topics in scientific literature," *2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, USA, pp. 1694-1697, 2017.

[8] W. Lu, S. Huang, J. Yang, Y. Bu, Q. Cheng and Y. Huang, "Detecting research topic trends by author-defined keyword frequency," *Information Processing & Management*, vol. 58, no.4:102594, 2021.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[10] W. Lou and J. Qiu, "Semantic information retrieval research based on co-occurrence analysis," *Online Information Review*, vol. 38, no. 1, pp. 4-23, 2014.

[11] L. C. Freeman, "Centrality in social networks: Conceptual clarification, *Social network: critical concepts in sociology. Londres: Routledge*, vol. 1, pp. 238-263, 2002.

[12] P. Bonacich, "Technique for analyzing overlapping memberships," *Sociological methodology*, vol. 4, pp. 176-185, 1972.

[13] M. E. J. Newman, "A measure of betweenness centrality based on random walks," Social networks, vol. 27, no. 1, pp. 39-54, January 2005.

[14] A. N. Langville and C. D. Meyer, "A survey of eigenvector methods for web information retrieval," *SIAM review*, vol. 47, no. 1, pp. 135-161, 2005.

[15] United States Patent and Trademark Office. (n.d.). Bulk Data Storage System (BDSS) Version 2.0.0. Retrieved January 12, 2022, from `https://bulkdata.uspto.gov/`

[16] United States Patent and Trademark Office. (n.d.-b). Classification Resources. Retrieved May 4, 2022, from `https://www.uspto.gov/web/patents/classification/cpc/html/cpc.html`

[17] J. E. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," in *Proceedings of the first instructional conference on machine learning*, vol. 242, no. 1, pp. 29-48, December 2003.

[18] A. J. C. Trappey, C. V. Trappey and C. Y. Wu, "Automatic patent document summarization for collaborative knowledge systems and services," *Journal of Systems Science and Systems Engineering*, vol. 18, pp. 71-94, 2009.

[19] A. Landherr, B. Friedl and J. Heidemann, A critical review of centrality measures in social networks," *Business & Information Systems Engineering*, vol. 2, pp. 371-385, October 2010.

[20] E. J. Yates and L. C. Dixon, "PageRank as a method to rank biomedical literature by importance," *Source code for biology and medicine*, vol. 10, no. 1, pp. 1-9, 2015.

[21] P. Branco, L. Torgo and R. P. Ribeiro, "SMOGN: a Pre-processing Approach for Imbalanced Regression," *Proceedings of Machine Learning Research*, vol. 74, pp. 36-50, 2017.

[22] *Mean Squared Error*, The Concise Encyclopedia of Statistics, New York, Springer, 2008, pp. 337–339.

[23] T. Gneiting and A. E. Raftery, "Strictly Proper Scoring Rules, Prediction, and Estimation," *Journal of the American statistical Association*, vol. 102, no. 477, pp. 359-378, 2007.

[24] L. Torgo, R. P. Ribeiro, B. Pfahringer and P. Branco, "SMOTE for regression," in *Progress in Artificial Intelligence*, pp. 378-389, September 2013.

[25] F. Pedregosa *el at.* , "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[26] A. Hagberg, P. J. Swart and D. A. Schult, "Exploring network structure, dynamics, and function using NetworkX," *SCIPY 08*, no. LA-UR-08-05495; LA-UR-08-5495). Los Alamos National Lab.(LANL), Los Alamos, United States, 2008.

[27] N. Kunz, (2020). SMOGN: Synthetic Minority Over-Sampling Technique for Regression with Gaussian Noise (Version v0.1.2). Retrieved from `https://pypi.org/project/smogn/`

[28] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature," *Geoscientific model development*, vol. 7, no. 3, pp. 1247-1250, 2014

[29] C. Lewis-Beck and M. Lewis-Beck, "Applied regression: An introduction," *Sage publications*, vol. 22, 2015.

[30] J. R. Taylor and W. Thompson, "An introduction to error analysis: the study of uncertainties in physical measurements, *University science books*, vol. 2, pp. 193-200, Mill Valley, CA, 1982.

[31] C. Goutte and E. Gaussier, "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation," *Advances in Information Retrieval*, Springer, Berlin, Heidelberg, pp. 345–359, 2005.

[32] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint*, arXiv:1212.5701, 2012.

[33] F. Chollet, "Keras," GitHub. Retrieved 2015, from `https://github.com/fchollet/keras`.

[34] J. Choi and Y.-S. Hwang, "Patent keyword network analysis for improving technology development efficiency," *Technological Forecasting and Social Change*, vol. 83, pp. 170-182, March 2014.

Krisanapong Yodprchavigit received the B.Eng. degree in Information and Communication Engineering from International School of Engineering, Chulalongkorn University, Thailand, in 2020. He is currently pursuing M.Sc. degree in Computer Science from Chulalongkorn University, Thailand.

Sukree Sinthupinyo received the B.Eng., M.Eng., and Ph.D. degrees in computer engineering from Chulalongkorn University, Thailand, in 1995, 1999, and 2001, respectively. He is currently an Assistant Professor with the Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Thailand. His research interests include social network analysis and machine learning.