IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 9, 2019, accepted January 12, 2020, date of publication January 20, 2020, date of current version January 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2967817

Patent Analytic Citation-Based VSM: Challenges and Applications

JIAJU QI[1], LEI LEI[2], (Senior Member, IEEE), KAN ZHENG[1], (Senior Member, IEEE), AND XIANBIN WANG[3], (Fellow, IEEE)

[1]Intelligent Computing and Communication (IC^2) Lab, Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]University of Guelph, Guelph, ON, Canada
[3]Department of Electrical and Computer Engineering, The University of Western Ontario, London, ON N6A 5B9, Canada

Corresponding author: Kan Zheng (zkan@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61731004, and in part by the Beijing University of Posts and Telecommunications (BUPT) Excellent Ph.D. Students Foundation under Grant XTCX201828.

ABSTRACT Patent citations are significant components of patents, which play a vital role in the implementation of patent analysis. However, most of the existed models only focus on the text of patents and do not realize that citations can remedy missing information in the text. A method for citation modeling in patent analysis is proposed to generate patent citation trees in this paper. Correspondingly, a specific neural network is designed for extracting abstract features in patent citation trees. Then, on the basis of extracted features, a new citation-based vector space model (CVSM) combining citations with text of the patent database is constructed for the subsequent applications. An experiment is conducted based on real patents of USPTO. The experimental results show that the proposed CVSM has good performances in several applications, which demonstrate the effectiveness of the proposed CVSM.

INDEX TERMS Citation modeling, neural networks, patent analysis, VSM.

I. INTRODUCTION

Recently, the number of patents is growing rapidly with the development of science and technology. Patents are one of the most effective indicators to evaluate the current development status of Intellectual Property Rights (IPR) [1]–[3]. Patent analysis results can tell research engineers how to identify current technology hotspots, predict the technology development trend [4]–[6] and guide research and development (R&D) project [7]. However, the huge number of patents makes manual patent analysis a very complicated and time-consuming task. The use of legal language and specific consideration of IPR definition makes patent text obscure and not easy to understand even for professionals in the related technical areas [8], which dramatically increases the difficulty of manual analysis. Several conventional methods of using keywords can be used to improve patent search [9], [10] or classification [11], and deal with the problem of huge data to some extent [12], [13], the results are not satisfying due to the limited features reflected by the keywords.

The associate editor coordinating the review of this manuscript and approving it for publication was Ah Hwee Tan.

Under this background, some automated patent analysis models based on the analysis of patent text have been proposed in recent years [14]–[16]. These models focused on text representation and semantic understanding. Some early models use traditional natural language processing (NLP) methods to construct vector space models (VSM) and embed a group of patents into a vector space, such as term frequency-inverse document frequency (TF-IDF) in [17]. Moreover, machine learning can be applied to extract complex and abstract features in patent text, such as convolution neural networks (CNN) in [18]. These models have improved the accuracy of patent analysis.

It should be noted that there are various types of data provided by a patent besides patent text, which can be divided into two parts, i.e., **unstructured** data (such as title, abstract, description, claims) and **structured** data (such as filed dates, inventors' names, International Patent Classification (IPC) codes, citations) [19], [20]. By analyzing such comprehensive data, useful features can be abstracted for developing a more effective model. Among the above data, patent citations are extremely important in the implementation of patent analysis. One objective for patent citations is to distinguish the claimed invention from existing inventions. The other objective is to provide background information. Citations can tell researchers what areas of technologies the claimed invention might involve [21]. Such citation data reflects the relationship among the related patents and is very helpful for patent analysis by clustering the patents and understanding the history of given patents [22], [23]. Moreover, citations can remedy missing information in the text. Thus, the analysis of citations provide an important aid to the analysis of text. However, most of the existed patent analysis models only exploit the features based on the patent text without making use of patent citations. As far as we know, there are very few methods for patent citation modeling.

In this paper, a method for citation modeling is proposed to generate patent citation trees. These citation trees as the enhanced patent representation combine citation data with text data. Correspondingly, a specific neural network, called as citation trees convolution neural networks (**CCNN**), is designed for mining abstract features from patent citation trees. Based on the well-trained CCNN, a new VSM derived from the patent citation trees, called as citation-based VSM (**CVSM**), is constructed for subsequent applications. In the proposed CVSM, the features of patents are abstracted by mining both the text data and citation data of patents. Compared with the existed VSMs, CVSM can utilize information from not only an individual patent, but also the related ones. We conduct experiments based on the U.S. patent data, and design several applications, i.e., patent similarity comparison, patent clustering and patent map generation, for CVSM to validate the effectiveness of our proposed method.

The rest of the paper is organized as follows. In Section II, background work is summarized and the challenges about this research is expounded. Section III gives the method of citation modeling and the pre-processing step for citation trees. In Section IV, a detailed description of the proposed CCNN is provided to handle patent citation trees. In Section V, the experiment of real patent database is conducted as a case study, including collection and pre-processing of the data set, the construction of CVSM and several applications based on CVSM. Finally, the conclusion is drawn in Section VI.

II. BACKGROUND

In the earlier study, patent philology is the main research strategy for patent analysis, including document statistics and collation [1]. With the development of NLP, some traditional NLP algorithms were applied in patent analysis to mine patent text potential information. [24] extracted the significant and rare keywords by Term Frequency-Inverse Document Frequency (TF-IDF) from patent text. [25] used dependency relationships to perform semantic analysis. [26] applied pre-trained Latent Dirichlet Allocation (LDA) models and dependency trees to conduct patents prior-art search. [27] studied the LDA algorithm results for different classes of patents. [28] extracted the descriptions of sci-tech effects and morphological features based on TF-IDF and links between words. Subsequently, the rise of machine learning have an influence on patent analysis. [29] constructed classification system with support vector machine. With the advances of deep learning, CNN was applied in text analysis and showed good performances in sentence classification [30]. Then, various CNN models have been proposed to be applied to NLP and achieved good performances, such as DCNN [31], RCNN [32], CNN-RNN [33], etc. Thus, the VSM based on CNN, e.g., featured VSM (FVSM), was proposed for patent analysis in [18].

VSM is a very useful patent analysis method. VSM was first proposed for text modeling and information retrieval in 1975 [34]. Each document was computed and mapped into a feature vector space. A wide range of application of VSM is the similarity comparison, such as [35]–[37]. Based on the VSM, the degree of similarity can be directly calculated quantitatively by feature vectors. In addition, VSM was also applied for clustering and generating patent maps [38]. There are several ways to generate VSMs. For example, the VSM based on the IPC code was proposed in [39]. Another widely-used VSM was generated by TF-IDF in [17]. The feature vector of each patent was obtained by calculating the TF-IDF weight of each term. In this way, the entire United States Patent & Trademark Office (USPTO) patents were mapped into a single vector space. Another way to generate VSM is based on LDA. In [40], the TF-IDF vector of each patent was calculated so that the LDA algorithm was used to calculate the patent-topic vector of each patent.

So far, almost all the existed VSMs are only based on the text of patents. However, citation data is also an important component of patents [41], [42]. In [43], citations have been used as an important reference to quantify the degree of patent value. By studying the network formed by citations in specific technical fields, [23] and [44] respectively drew the development trajectory of the technology, as an important indicator of patent analysis. In [45], a loglinear relationship between patent citations and patent value was studied. It has been shown that patent citation data plays an important role in patent analysis applications, such as tracing technology development routes.

The citation data needs to be formulated before being utilized in VSM. In most cases, citations are modeled as networks or trees. For example, in order to study patents in the field of nanotechnology, the patents made up a huge network for patent analysis based on the relationship of citations [22]. In [46], a model for quantitative calculation of multi-stage citation data has been proposed and been used for simple clustering. It should be noted that different data structures of citations will lead to diverse design ideas for the following neural network. Considering that the tree structure is more concise and there exists mature processing models, citations are processed into citation trees.

Most of existed CNNs are used to process traditional data in the type of matrix, such as some classic structures [47]–[49]. For sake of processing data in the type of tree, a tree-based convolution neural network, i.e., tree-based convolution neural network (TBCNN), was proposed, which can process the programming language process based on triangular convolution kernels [50]. After being transformed to abstract syntax tree (AST) trees, programs can be classified to several categories. More works on TBCNNs have appeared, e.g., sentence modeling in [51], natural language inference and heuristic matching in [52]. Inspired by TBCNN, a neural network model of tree-based convolution for processing citation trees to extract the features of citation data is proposed in this paper.

III. CITATION MODELING BASED ON PATENT DATABASE

A. DESIGN OF PATENT CITATION TREES

It is necessary to observe real patent data for the appropriate citation modeling. Most of the citation data includes two kinds of citations, i.e., the citations by **patents** and **non-patents** such as papers, protocols, etc. The former is also called as citation patents, whose format is more normative than that of the latter. In essence, the latter can be regarded as the text data of patents, which has no specification and is more difficult to be dealt with than the former. Therefore, only the citation patents are focused on in our research. Furthermore, the citation patents may have their own citation patents. Based on such relationship, the data of a tree topology structure is suitable for citation modeling. In this way, patent citation trees can be established layer by layer. By analyzing patent citation trees, the meaningful patent analysis can be carried out including tracing the development history of patents, summarizing trends in technology, and so on.

The method of establishing patent citation trees can be expounded by the following structure model as an example: Assume that a specific **Patent A** cites two patents, i.e., **Patent B** and **Patent C**. **Patent B** cites **Patent D** and **Patent E**. **Patent C** cites **Patent F**. The corresponding patent citation tree of **Patent A** can be constructed as shown in Fig. 1. This citation tree consists of three layers. The root node of the tree, i.e., **Patent A**, is to be analyzed in the following. Patents in the second layer of the tree are citations of **Patent A**, namely **First Level Citations**. Patents in the third layer of the tree are citations of First Level Citations, namely **Second Level Citations**. By this way, patent citation trees with several layers can be established. However, considering that the patents across more than two layers have not strong connection, we discard patents after Second Level Citations and only keep the patent citation trees of three layers.

B. REVIEW ON FVSM

The modeling rule above only clarifies the topological structure of patent citations. Next, we should establish a quantitative model representation. Each node in the tree stores the origin feature vector of its corresponding patent. These origin feature vectors can be generated by FVSM [18] or other VSMs based on the analysis result of patent text. FVSM is a feature vector space model of the patent database, which is constructed by feature vectors extracted by a machine learning method. Neural networks have strong ability to mine abstract semantic features from patent text. Compared with VSMs based on traditional NLP methods, FVSM can analyze more sentence-level semantics, which is more complex, comprehensive and high-level than word-level semantics. Meanwhile, by embedding patents into a vector space of the appropriate size, complex patent analysis tasks can transform to simple mathematical operations.

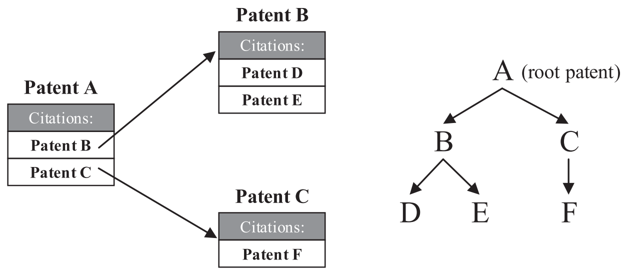


FIGURE 1. An example for the topological structure of patent citation trees.

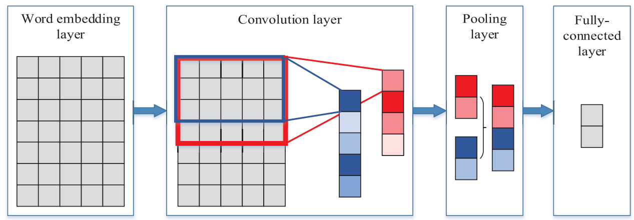


FIGURE 2. The four-layers structure of CNN used for the construction of FVSM.

In practical terms, since CNN has various advantages, including simple structures, easily training, convenient deployment, flexible scalability and good performance, CNN is chosen to extract features from patent text in our work. Concretely speaking, the structure of CNN consists of four layers, as shown in Fig. 2. The first layer is word embedding layer, which is used to embed words into a vector space model. In this way, patent text can be quantitatively represented as matrices as the input of convolution. The second layer is convolution layer. The sizes of convolutional kernels are from three to five. For each size, the number of kernels is 100. Kernels slide along the matrix and capture semantic features from words in the sliding window. The third layer is pooling layer, where maximum pooling is implemented as the pooling strategy. The aim is to acquire the best pattern matching. The last layer is fully-connected layer used for patent classification as the training task. After well-traing CNN, the output of pooling layer can be regarded as the features mined from patent text. Based on these features, FVSM can be constructed as the origin feature vectors, in other words, the raw materials of patent citation trees.

C. FURTHER PROCESSING OF PATENT CITATION TREES

Considering that neural networks can extract abstract and connotative features contained in the patent citation trees, neural networks are introduced in our research to handle citation trees. A specific neural network needs to be designed for analyzing the tree structure of the patent citation data.

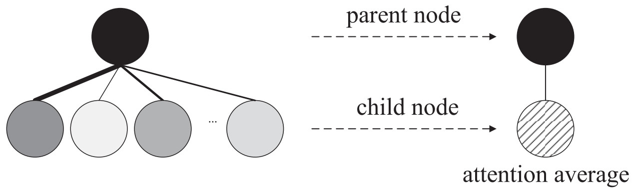


FIGURE 3. Pre-processing method for a subtree. "Attention average" all the child nodes into one value. The darker the color of the child node, the bigger its corresponding weight.

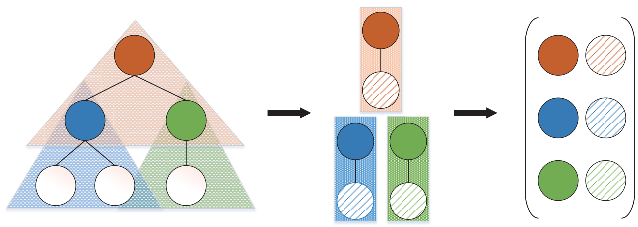


FIGURE 4. The way of converting a citation tree to a matrix. The citation tree with tree subtrees is transformed into the matrix with tree rows.

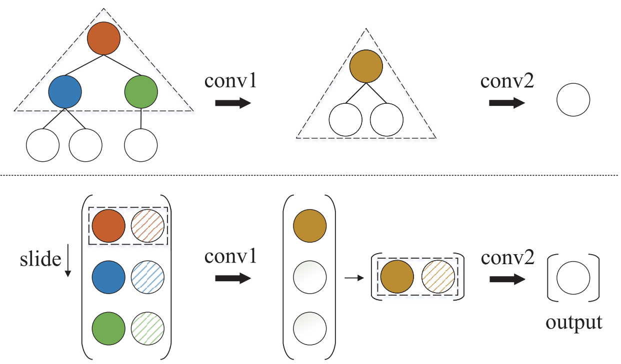


FIGURE 5. Tree-based convolution for patent citation trees. By the way of the process from trees to matrices, tree-based convolution is equivalent to matrix-based convolution.

Tree-based convolution kernels are suitable for the tree structure, which are applied in the neural networks to extract the features contained in patent citation trees.

Since patent citation trees have a kaleidoscopic structure, they are not available as the input of neural networks directly. These citation trees need to be further processed into the canonical form of matrices. A subtree is the basic unit of patent citation trees. Fig. 3 presents the pre-processing method for a subtree with one parent node. Inspired by the thoughts of attention which is firstly applied in [53] in NLP, a strategy namely "attention average" is proposed to process a subtree. The parent node can be represented as $\boldsymbol{v_p}$ and child nodes can be represented as $\boldsymbol{v_1, v_2, \ldots, v_i, \ldots, v_n}$.

1) Calculate the match scores between the parent node and each child node. There are several ways to match two vectors, such as inner product, cosine similarity and Euclidean distance.

$$a_i = Score\left(v_p, v_i\right)$$

2) Conduct softmax on the match scores to generate the weight.

$$a_i' = Soft\max\left(a_i\right)$$

3) Weighted average all the child nodes and generate the new child node $\boldsymbol{v_c}$.

$$v_c = \sum_{i=1}^{n} a_i' v_i$$

The design of "attention average" shows the difference between citations. The more similar the citation, the greater its weight. By the strategy of "attention average", the subtree changes to a normative subtree with one parent node and one child node. Considering that the number of citations is uncertain, the method of "attention average", i.e., weighted averaging all the child nodes, can make our pre-processing method general for subtrees with any number of child nodes. Then, when designing neural networks, in each convolution kernel one value is assigned for the weighted average value of child nodes and the other one is assigned for the parent node.

By the above rules, each patent citation tree can be represented by a matrix as shown in Fig. 4. The root node in the first level and nodes in the second level form the first subtree. Then, the first subtree composes the first row of the matrix. Two nodes in the second level and their corresponding nodes in the third level form the second and the third subtree respectively. Next, they compose the second and the third row of the matrix. This matrix has the following properties, i.e.,

1) The number of rows in the matrix is the number of First Level Citations plus one.
2) Values in the first row represent the subtree consisting of the root patent and its First Level Citations, which is important for designing the neural network.
3) Each row represents a subtree in the citation tree. In each row, the first value represents the parent node and the second value represents the attention average value of child nodes.

IV. ARCHITECTURE OF CITATION TREES CONVOLUTION NEURAL NETWORKS (CCNN)

A. DESIGN OF TREE-BASED CONVOLUTION KERNELS FOR CCNN

After transformed to matrices, patent citation trees can be the input of convolution neural networks. Fig. 5 illustrates the tree-based convolution with the patent citation tree as shown in Fig. 1. The size of kernels is 1×2. For each kernel, the first value needs to multiply values representing parent nodes and the second value need to multiply values representing the attention average of child nodes. One dimensional convolution is applied here, which means the width of the convolution kernel is the same as the width of the input matrix. Kernels slide along the longitudinal direction of the matrix as sliding windows and employ convolution with each row, which is called as the first convolution, i.e., Conv1. The resulting output of Conv1 includes three values. The first one corresponds to the first subtree (consisting of the root

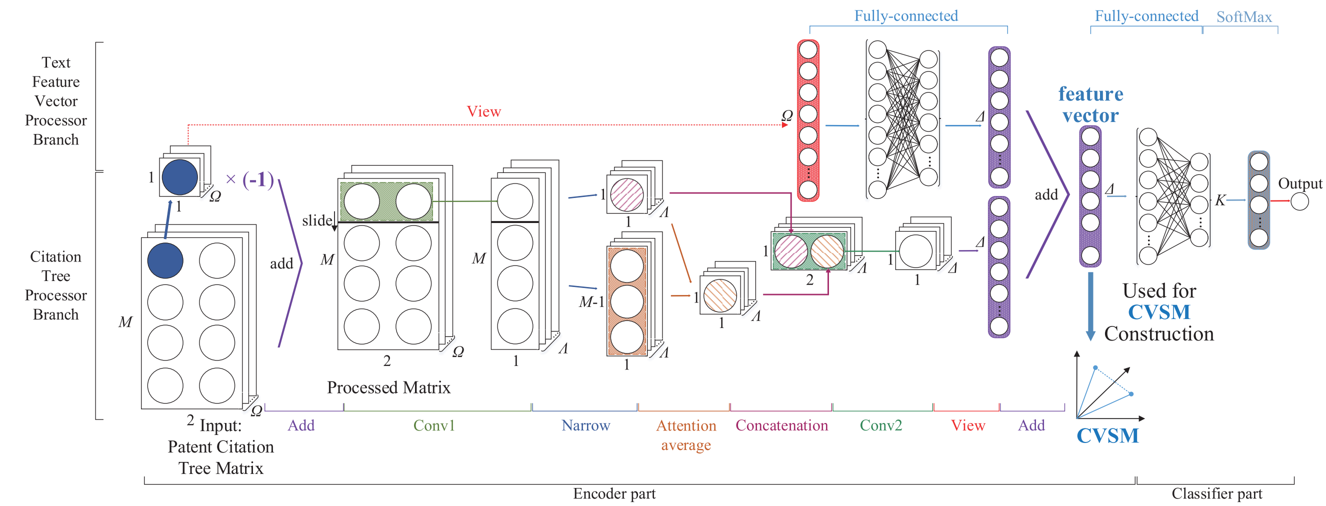


FIGURE 6. Detailed architecture of CCNN.

patent and its citation patents) in the citation tree. The second one and the third one correspond to the second and the third subtree in the citation tree, respectively. Thus, the three values as the result of Conv1 are hierarchical and compose a new tree consisting of one parent node and two child nodes. The new subtree can be converted to a 1×2 matrix as the input data of the next convolution. The process of the second convolution (Conv2) is the same as Conv1. Finally, the result from the output of Conv2 can be obtained as the abstract feature of the patent citation tree extracted by the neural network.

For citation trees with different layers, the numbers of convolutions are different, which equal to the number of layers minus one correspondingly. For example, if the citation trees have three layers, two convolutions should be performed to extract the features. Nevertheless, the method of each convolution and the process from trees to matrices are kept same.

B. DETAILED STRUCTURE OF CCNN

In order to implement the above convolution method, CCNN is designed for processing patent citation trees. As shown in Fig. 6, CCNN mainly consists of two parts, i.e., the **encoder** and the **classifier**. The former extracts the abstract features of patent citation trees while the latter is used for patent classification. In addition, there are two branches in the encoder, i.e., **text feature vector processor branch** and **citation tree processor branch**. The details are given as follows, i.e.,

1) ENCODER: CITATION TREE PROCESSOR BRANCH

a: MATRIX PRE-PROCESSING

The essence of the proposed CCNN is a function that makes mathematical transform based on feature vectors of the specific patent to be analyzed and its citations. The function adjusts the feature vector of the specific patent through analyzing citations and figure out the new one. We consider the differences between the feature vectors of citations

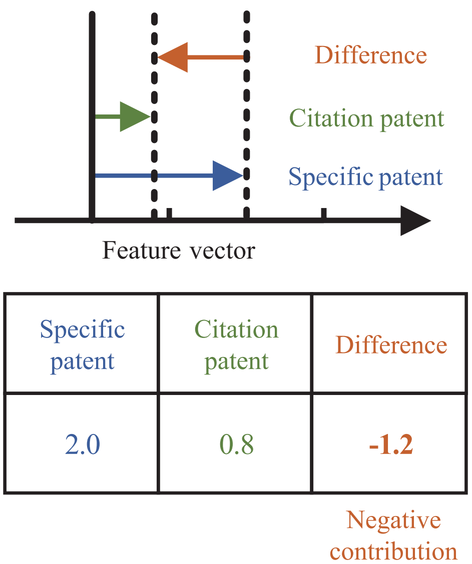


FIGURE 7. An example for the influence of difference between the citation patent and specific patent.

and the one of the specific patent have influence on the adjustment of feature vectors. The following example shown in Fig. 7 can illustrate this point. Assuming that features are one-dimensional vector, it is found that the feature of citation patent is smaller than the one of specific patent. Therefore, the difference gives a negative contribution. In this way, the new feature vector of the specific patent should be smaller than the old one.

The output of the pre-processing method for citation trees, i.e., patent citation tree matrices, is three-dimensional matrices. Each node in citation trees stores the feature vector of FVSM, i.e., the text feature vector. Matrix pre-processing should be performed to figure out the processed matrix. The matrix $\boldsymbol{T}$ denotes the patent citation tree matrix, i.e., $\boldsymbol{T} \in \mathbb{R}^{M \times N \times \Omega}$. M denotes the number of subtrees which can be returned by splitting the patent citation tree. N is a constant of 2 representing the parent node and the attention average of child nodes in a subtree. Ω denotes the dimension of vectors in FVSM, i.e., the number of feature maps in $\boldsymbol{T}$.

The first step is extracting the text feature vector $\boldsymbol{F} \in \mathbb{R}^{1\times 1\times \Omega}$ of the specific patent, which is represented by the root node of the citation tree. Therefore, $\boldsymbol{F}$ can be expressed by:

$$F_{1,1,\omega} = T_{1,1,\omega}, (\omega \in \{1,\ldots,\Omega\}). \tag{1}$$

The second step is the generation of the processed matrix. Let the matrix $\boldsymbol{X}$ represent the processed matrix with three dimensions, i.e., $\boldsymbol{X} \in \mathbb{R}^{M\times N\times \Omega}$. For each feature map, the processed matrix $\boldsymbol{X}$ can be calculated as follows:

$$X_{m,i,\omega} = T_{m,i,\omega} - F_{1,1,\omega}, \tag{2}$$

where $X_{m,i,\omega}$ and $T_{m,i,\omega}$ denotes the value of the m-th row ($m \in \{1,\ldots,M\}$), the i-th column ($i \in \{1,2\}$) of the ω-th feature map ($\omega \in \{1,\ldots,\Omega\}$) in $\boldsymbol{X}$ and $\boldsymbol{T}$.

b: CONV1

Two convolution layers are the cores of the proposed CCNN. The matrix $\boldsymbol{X}$ representing the processed matrix can be as the input to the convolution layer. The purpose of Conv1 is to mine semantic information and inheritance relationship in patent citation trees, then obtain new feature trees with less layers and more concise structures. In Fig. 6, the kernels in Conv1 have the size of 1×2, i.e., filters slide through every $(M \times 2)$-dimensional feature map along its longitudinal direction instead of along both longitudinal and transverse directions. The number of kernels is set to be Λ. Let $\boldsymbol{W_{0,\lambda}} \in \mathbb{R}^{1\times 2}$ denote the weight of the λ-th kernel ($\lambda \in \{1,\ldots,\Lambda\}$) and b_0 denote the bias. For the block of matrix $\boldsymbol{X}$ in the m-th row ($m \in \{1,\ldots,M\}$), the feature value $c_{m,\lambda}$ can be calculated as follows:

$$c_{m,\lambda} = \sigma\left(\sum_{i=1}^{2}\sum_{\omega=1}^{\Omega} W_{0,\lambda_{(1,i)}} X_{m,i,\omega} + b_0\right), \tag{3}$$

where $W_{0,\lambda_{(1,i)}}$ denotes the i-th ($i \in \{1,2\}$) value in $\boldsymbol{W_{0,\lambda}}$, $X_{m,i,\omega}$ denotes the value of the m-th row, the i-th column of the ω-th feature map ($\omega \in \{1,\ldots,\Omega\}$) in $\boldsymbol{X}$, and σ denotes the activation functions, i.e., Sigmoid, Tanh, ReLU, etc. In this way, for the λ-th kernel, the feature vector $\boldsymbol{c_\lambda} \in \mathbb{R}^{M\times 1}$ can be generated as $\boldsymbol{c_\lambda} = \left[c_{1,\lambda}; c_{2,\lambda}; \ldots; c_{m,\lambda}\ldots; c_{M,\lambda}\right]$. The output of Conv1, namely $\boldsymbol{c} \in \mathbb{R}^{M\times 1\times \Lambda}$, can be generated as $\boldsymbol{c} = [\boldsymbol{c_1}, \boldsymbol{c_2}, \ldots, \boldsymbol{c_\lambda}, \ldots, \boldsymbol{c_\Lambda}]$, where Λ denotes the number of future maps in $\boldsymbol{c}$.

c: TNSOR ORGANIZING

The output $\boldsymbol{c}$ of Conv1 cannot be the input of Conv2 directly. Therefore, the purpose of tensor organizing is transforming $\boldsymbol{c}$ into the available input data of Conv2, i.e., transforming the output of Conv1 into a new tree by the way in Fig. 5 and converting the new tree to the matrix by the pre-processing method in Fig. 4. The narrow layer, attention average layer and concatenation layer are used to achieve the procedure of tensor organizing.

In the narrow layer, $\boldsymbol{c}$ is divided into two parts, where the first part $\boldsymbol{c_p} \in \mathbb{R}^{1\times 1\times \Lambda}$ represents the parent node in the new tree, while the second part $\boldsymbol{c_c} \in \mathbb{R}^{(M-1)\times 1\times \Lambda}$ represents the child nodes in the new tree. The separate processing of these two parts also divides the neural network into two branches. The first branch corresponding to $\boldsymbol{c_p}$ has no follow-up operation. While the second branch corresponding to $\boldsymbol{c_c}$ should take the attention average of $\boldsymbol{c_c}$ based on its first dimension by calculating match scores with $\boldsymbol{c_p}$. The output of the attention average layer is $\boldsymbol{c'_c} \in \mathbb{R}^{1\times 1\times \Lambda}$. It can be found that the dimensions of two resulting outputs $\boldsymbol{c_p}$ and $\boldsymbol{c'_c}$ of two branches are identical.

In the concatenation layer, $\boldsymbol{c_p}$ and $\boldsymbol{c'_c}$ merge together based on their second dimension. Then, the output is $\boldsymbol{H} \in \mathbb{R}^{1\times 2\times \Lambda}$ and $\boldsymbol{H} = \left[\boldsymbol{c_p}, \boldsymbol{c'_c}\right]$. In this way, the process of the new tree is completed. $\boldsymbol{H}$ can be used as the input of Conv2.

d: CONV2

Similar to Conv1, the purpose of Conv2 is to mine deeper semantic information and inheritance relationship in patent citation trees, then retrieve the abstract feature of patent citation trees as the results. In Fig. 6, the kernels in Conv2 also have the size of 1×2. The number of kernels is set to be Δ. $\boldsymbol{W_{1,\delta}} \in \mathbb{R}^{1\times 2}$ represents the weight of the δ-th kernel ($\delta \in \{1,\ldots,\Delta\}$) and b_1 is the bias. For matrix $\boldsymbol{H}$, the feature value d_δ can be calculated as follows:

$$d_\delta = \sigma\left(\sum_{i=1}^{2}\sum_{\lambda=1}^{\Lambda} W_{1,\delta_{(1,i)}} H_{1,i,\lambda} + b_1\right), \tag{4}$$

where $W_{1,\delta_{(1,i)}}$ denotes the i-th ($i \in \{1,2\}$) value in $\boldsymbol{W_{1,\delta}}$, $H_{1,i,\lambda}$ denotes the value of the first row, the i-th column of the λ-th feature map in $\boldsymbol{H}$, and σ also denotes the activation functions. In this way, the output of Conv2, namely $\boldsymbol{d} \in \mathbb{R}^{1\times 1\times \Delta}$, can be generated as $\boldsymbol{d} = [d_1, d_2, \cdots, d_\delta, \ldots, d_\Delta]$, where Δ denotes the number of future maps in $\boldsymbol{d}$.

2) ENCODER: TEXT FEATURE VECTOR PROCESSOR BRANCH

The citation tree processor branch aims to extract abstract features based on the difference between the specific patent and its citations. Therefore, the above features should add on the original features, i.e., the feature vector of the specific patent. The whole process is similar to ResNet [54].

In order to make the dimensionality of the above two feature vectors consistent, a fully-connected layer is used in this branch, where the weight is $\boldsymbol{W_2} \in \mathbb{R}^{\Delta\times \Omega}$ and the bias is $\boldsymbol{b_2} \in \mathbb{R}^{\Delta\times 1}$. The input of this layer is a one-dimensional vector $\boldsymbol{f} \in \mathbb{R}^{\Omega\times 1}$, which is resized from $\boldsymbol{F}$. The output $\boldsymbol{f'} \in \mathbb{R}^{\Delta\times 1}$ can be calculated as follows:

$$\boldsymbol{f'} = \boldsymbol{W_2} \times \boldsymbol{f} + \boldsymbol{b_2}. \tag{5}$$

The last step of the encoder is combining the features of two branches, i.e., $\boldsymbol{d'}$ and $\boldsymbol{f'}$, where $\boldsymbol{d'} \in \mathbb{R}^{\Delta\times 1}$ is the one-dimensional vector resized from $\boldsymbol{d}$. Then, the feature vector $\hat{\boldsymbol{f}}$ from CCNN is obtained. $\hat{\boldsymbol{f}}$ can be calculated as follows:

$$\hat{\boldsymbol{f}} = \boldsymbol{d}' + \boldsymbol{f}'. \tag{6}$$

3) CLASSIFIER

After the encoder, the classifier is used in the neural network for patent classification. After training the neural network, the accuracy of patent classification can be calculated as the quantitative index for evaluating CCNN. The input data of the classifier is the output data $\hat{\boldsymbol{f}} \in \mathbb{R}^{\Delta \times 1}$ of the encoder. The classifier includes two layers, i.e., the fully-connected layer and the softmax layer.

$\hat{\boldsymbol{f}}$ is input to the fully-connected layer, where the weight is $\boldsymbol{W_3} \in \mathbb{R}^{K \times \Delta}$ and the bias is $\boldsymbol{b_3} \in \mathbb{R}^{K \times 1}$. The output $\boldsymbol{y} \in \mathbb{R}^{K \times 1}$ of this layer is

$$\boldsymbol{y} = \boldsymbol{W_3} \times \hat{\boldsymbol{f}} + \boldsymbol{b_3}. \tag{7}$$

Finally, $\boldsymbol{y}$ is input to the softmax layer. The resulting output $\hat{\boldsymbol{y}}$ of this layer is the probability that the patent belongs to each category. The maximum value of $\hat{\boldsymbol{y}}$ is selected for the output of CCNN as the classification result, i.e.,

$$\hat{\boldsymbol{y}} = Softmax\left(\boldsymbol{y}\right). \tag{8}$$

Since the architecture of the proposed CCNN adopts the idea of taking the attention average of child nodes in each subtree, CCNN has the ability of handling any type of three-layers tree. For any specific patent, the corresponding matrix input into the neural network is $\boldsymbol{T} \in \mathbb{R}^{M \times N \times \Omega}$, where the value of N and Ω are required to be consistent with all the patents, and the value of M is not fixed to patents.

C. CONSTRUCTION OF CVSM BASED ON WELL-TRAINED CCNN

The output $\hat{\boldsymbol{f}}$ of the encoder is the abstract feature extracted by CCNN. This new feature is obtained by analyzing citation trees based on FVSM. In this way, after training CCNN and collecting new features of all the patent citation trees in the data set, a new VSM, i.e., CVSM, can be constructed for further patent analysis.

V. APPLICATIONS OF CVSM

In order to validate the effectiveness of CCNN on patent analysis, various experiments are conducted based on the data set composed of real patents. Several application scenarios are implemented as the case study in our work.

A. CONSTRUCTION OF EXPERIMENT DATA SET

The first step of the experiment is data set construction, which can be divided into two steps, i.e., collection and pre-processing of experiment data set.

1) COLLECTION OF EXPERIMENT DATA SET

Our experiments are carrrried out on the data set of U.S. patents. The data set includes the patents in USPTO from the year 2015 to 2017 under the IPC class of H04 (representing electric communication technique), namely as $\mathbb{S}_A$ [55].

References Cited [Referenced By]

U.S. Patent Documents (reserved)

4640939	February 1987	Cavezzan et al.
4670531	June 1987	Eckberg
9120935	September 2015	Marrot et al.
2009/0288431	November 2009	Rao
2012/0328787	December 2012	Marrot et al.
2014/0004359	January 2014	Marrot

Foreign Patent Documents (discarded)

0238033	Sep 1987	EP
0238033	Sep 1987	EP
9210529	Jun 1992	WO
2011076710	Jun 2011	WO
2012085364	Jun 2012	WO
2012175825	Dec 2012	WO

Other References (discarded)

Kiyohiro et al, EP 0238033 Machine Translation, Sep. 23, 1987. cited by examiner .
International Search Report dated Jan. 28, 2015 in counterpart European Application No. PCT/EP2014/003352. cited by applicant.

Primary Examiner: Whiteley; Jessica E
Attorney, Agent or Firm: McBee Moore Woodward & Vanik IP, LLC

FIGURE 8. An example of citations in a USPTO patent (US 9834648). Only the first kind of citations are reserved for further experiment.

Fig. 8 gives an example of the citation information of the USPTO patent with the number of US 9834648 [56]. There are three kinds of citation information, i.e.,

1) USPTO patents cited by the patent of US 9834648: Both the patent and its citation patents belong to USPTO.
2) Foreign citation patents: They cannot be linked and collected information directly.
3) Other types of citations instead of patents.

Since only the first kind of citations can be conveniently collected, they are reserved for the construction of patent citation trees in our experiment for the sake of analysis. Moreover, for each patent in $\mathbb{S}_A$, its valid patent citation tree needs to be constructed. The valid citation tree should satisfy the following conditions. For every patent in the citation tree, it should be discarded if it does not belong to $\mathbb{S}_A$. In other words, we only keep the citation patents that belong to $\mathbb{S}_A$. In this way, there are 4445 patents from $\mathbb{S}_A$ construct their own valid patent citation trees. These valid citation trees compose our experimental patent data set, namely $\mathbb{S}_B$. For the sake of the illustration, Fig. 9 gives an example to explain how to construct $\mathbb{S}_B$. **Patent B** is the citation of **Patent A**, and it belongs to $\mathbb{S}_A$ as well as **Patent A**. Hence, **Patent A** can construct a valid patent citation tree containing **Patent B**. As a result, **Patent A** can be used as a valid experimental record in $\mathbb{S}_B$. Conversely, **Patent C** only has a citation namely **Patent D** which does not belong to $\mathbb{S}_A$. Thus, **Patent C** cannot construct a valid citation tree. As a result, **Patent C** is not contained in $\mathbb{S}_B$. In this way, old citation patents and citation

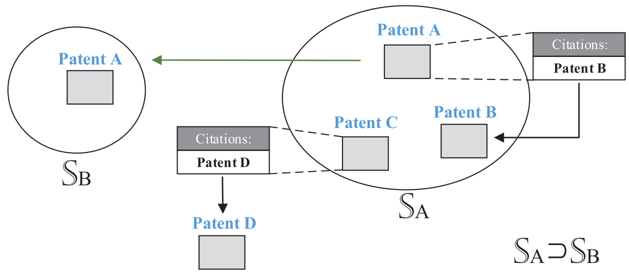


FIGURE 9. The generation of the data set $\mathbb{S}_B$. $\mathbb{S}_B$ is a subset of $\mathbb{S}_A$.

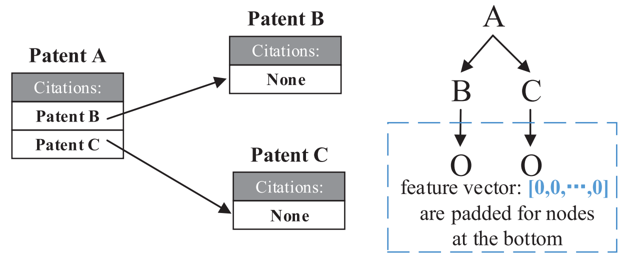


FIGURE 10. Details of "padding", which means padding the virtual patents with feature vectors of 0.

patents with huge difference in technical fields are filtered out.

2) PRE-PROCESSING OF EXPERIMENT DATA SET

After removing stop words and Porter stemming [57] for the rest of words, the FVSM of patents in $\mathbb{S}_A$ is constructed by the well-trained CNN. The sizes of convolution kernels are from three to five. The number of each size is 100. Thus, the dimension of FVSM in our experiment is 300.

When training CNN, we have to get the labels for patent classification. Due to the large numbers, patents could not be manual labeled directly. The work of manual labeling is hard and time-consuming. Besides, the purpose of the experiment is not only to validate the effectiveness of our model, but also perform patent analysis based on the real patent data, which belongs to an industrial application of technology. Thus, we hope our model have the ability of universal. In this way, a type of "half-automatic and half-manual" method is used in our work. An unsupervised NLP algorithm namely Latent Dirichlet Allocation (LDA) [58] is chosen for automatic labeling firstly. Eight topics are extracted by LDA based on the text data of patents in $\mathbb{S}_A$, corresponding to eight categories for classification. For each patent, the probabilities of the eight categories should be calculated and the category corresponding to the highest probability should be selected as the result of classification. In order to improve the accuracy of automatic labels, if a patent has some closest high probability values, it needs to be checked manually. This is why the method is called "half-automatic and half-manual".

Then, patent citation trees of patents in $\mathbb{S}_B$ can be constructed. Every node in trees stores the feature vector of FVSM of the corresponding patent. There are three layers in our citation trees, i.e., the specific patent, its First Level Citations and Second Level Citations. In the cases when there is no enough to build a citation tree with three layers, the "padding" operation is performed, which means padding the virtual patents with feature vectors of **0** for construction of the citation tree with three layers. The details of "padding" are shown in Fig. 10.

Finally, patent citation trees of patents in $\mathbb{S}_B$ are transformed to matrices as the input data of the following neural network, i.e., CCNN.

B. TRAINING CCNN AND CONSTRUCTING CVSM

In order to construct CVSM, the first step is training CCNN. The architecture of CCNN and the process of data flow is illuminated in Section IV. Fig. 11 shows the details of the applied CCNN in our experiment, including the parameter selection of the two convolution layers, the selection of feature vector's dimension, etc.

For citation tree processor branch, the number of input feature maps in Conv1 should equal to the dimension of FVSM, i.e., $\Omega = 300$. The number of outputting feature maps is set to be 100, i.e., $\Lambda = 100$. After Conv1, the output is $\boldsymbol{c} \in \mathbb{R}^{M\times1\times100}$. In the narrow layer, $\boldsymbol{c}$ is split into two parts $\boldsymbol{c_p}$ and $\boldsymbol{c_c}$ based on the second element of the first dimension of $\boldsymbol{c}$ as the boundary. In attention average layer, $\boldsymbol{c_c'}$ is obtained by taking the attention average of $\boldsymbol{c_c}$ based on its first dimension. Then, $\boldsymbol{c_p}$ and $\boldsymbol{c_c'}$ are combined to form the new matrix $\boldsymbol{H}$ in the concatenation layer for the input of Conv2. In Conv2, the number of input feature maps is $\Lambda = 100$. The number of output feature maps is set to be 30, i.e., $\Delta = 30$. For text feature vector processor branch, the weight of fully-connected layer is $\boldsymbol{W_2} \in \mathbb{R}^{30\times300}$. Therefore, we regard the output vector of the encoder, i.e.,$\hat{\boldsymbol{f}} \in \mathbb{R}^{30\times1}$, as the feature vector of the patent citation tree, in other words, the new feature vector of the patent to be analyzed.

In the classifier, the vector $\hat{\boldsymbol{f}}$ is input to the fully-connected layer with the weight of $\boldsymbol{W_3} \in \mathbb{R}^{8\times30}$. The output of this layer has 8 values. After the softmax layer they become the probabilities of 8 categories. In this way, patent classification has finished.

To be clear, the dimension of new feature vectors are reduced to 30 due to the following two reasons, i.e.,

1) The significance of VSM is to embed all the patents in a data set into a vector space. Patents in $\mathbb{S}_B$ for our experiment are selected in $\mathbb{S}_A$, so their number is far smaller than that of patents in the experiment of FVSM. Therefore, the dimension of the vector space of CVSM need not be set too large.
2) Larger dimension means more kernels, which can extract more features from patents. However, too large dimension could also bring the curse of dimensions [59]. If it happens, the data is more sparsely distributed in the vector space.

In order to obtain the scientific results, we adopted the method of 10-fold cross-validation [60] to train CCNN ten times. The meaning is to divide experimental data into ten samples. Each time of training CCNN, we select one sample as the test set for evaluation and the rest nine samples as

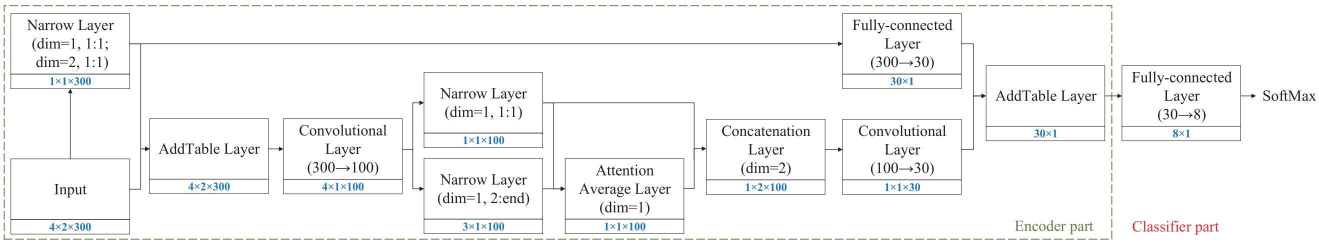


FIGURE 11. The detailed architecture of CCNN in the experiment. 30-dimensional feature vector is chosen. The numbers below each layer represent the dimension of output tensor for each layer.

TABLE 1. 10-fold cross-validation training results of CCNN. The number of iteration is 50.

Fold Index	1	2	3	4	5	6	7	8	9	10	Average
Train set Scores	98.88%	98.70%	98.67%	98.90%	98.70%	**98.97%**	98.74%	98.90%	98.92%	98.83%	98.82%
Test set Scores	99.30%	97.03%	94.24%	97.95%	98.85%	**99.77%**	98.40%	99.08%	98.85%	96.61%	98.01%

the train set for training. In the training process, random gradient descent algorithm with update rules of AdaDelta and a mini-batch size of 50 is used to train the neural network.

The training results of the proposed CCNN are shown in Table. 1. The number of iteration is 50. It can be seen that the average of train set scores can achieve 98.82%. In ten test set scores, the highest accuracy is 99.77%, the lowest is 94.24%, and the average is 98.01%. Experimental results prove that CCNN can well complete the task of patent classification.

After training CCNN, the results should be collected to construct CVSM for further patent analysis. The method of results collection can be divided into two steps, i.e.,

1) An appropriate CCNN model should be selected according to the test set scores in Table 1. The sixth model corresponding to the highest score is selected for the construction of CVSM. According to the architecture shown in Fig. 6, the part of the classifier is discarded and the well-trained parameters in two convolution layers, i.e., weight and bias, should be maintained.
2) Patent citation trees of patents in $\mathbb{S}_B$ are inputted into the above CCNN without the classifier again. The results are the output of encoder, which are 30-dimension feature vectors extracted from citation trees by CCNN.

In this way, CVSM is constituted based on these 30-dimension feature vectors.

C. PATENT SIMILARITY COMPARISON

Patent similarity is the typical application of the proposed CVSM. By mathematical calculation based on feature vectors of CVSM, the similarity of patents can be quantified. The data set for patent similarity is the patent triads consisting of $\{\mathbf{P}, \mathbf{P}^{+}, \mathbf{P}^{-}\}$. A patent triad contains three patents. The first one $\mathbf{P}$ is the specific patent as the datum and needs to be compared with the others. $\mathbf{P}^{+}$ is more similar than $\mathbf{P}^{-}$ compared with $\mathbf{P}$, which can be

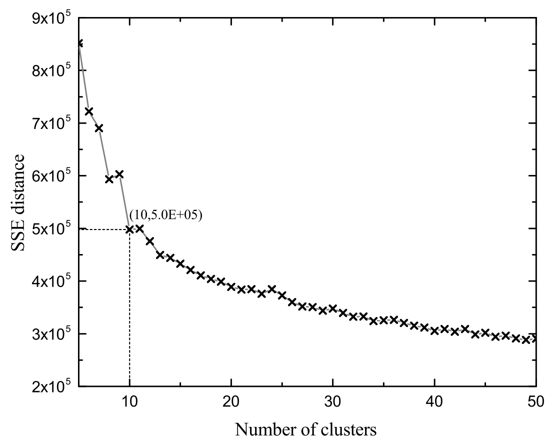
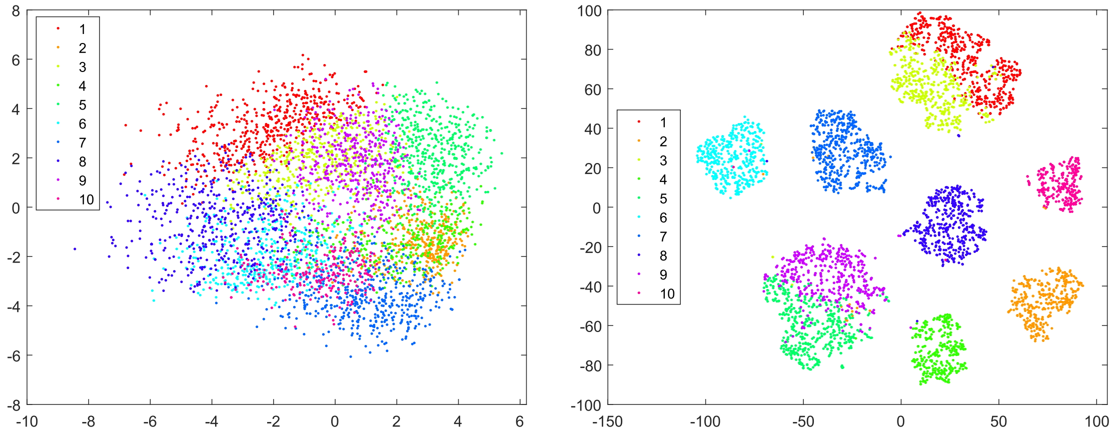


FIGURE 12. SSE distance of 5 to 50 clusters.

expressed as $similarity\,(P, P+) > similarity\,(P, P-)$, where $similarity\,(P, P+)$ denotes the quantized similarity between $\mathbf{P}$ and $\mathbf{P}^{+}$. There are several quantitative methods to calculate $similarity\,(P, P+)$ including Euclidean distance, i.e., $d\,(vec\,(P)\,, vec\,(P+))$, cosine of the angular separation, i.e., $\cos\,(\theta\,(vec\,(P)\,, vec\,(P+)))$ and Jaccard index [61], i.e., $J\,(vec\,(P)\,, vec\,(P+))$, where $vec\,(P)$ comes from CVSM or other VSMs. This is the same for $similarity\,(P, P-)$.

We manually label 300 groups of the above patent triads from $\mathbb{S}_B$ satisfying the conditions that the similarity between $\mathbf{P}^{+}$ and $\mathbf{P}$ is higher than that between $\mathbf{P}^{-}$ and $\mathbf{P}$. Meanwhile, the similarity between $\mathbf{P}^{+}$ and $\mathbf{P}$ and the one between $\mathbf{P}^{-}$ and $\mathbf{P}$ based on CVSM, CNN-based FVSM and TF-IDF-based VSM are calculated respectively inside each triad. Euclidean distance and Cosine similarity are chosen as the quantitative methods of similarity, i.e., $similarity\,(P, P+) = d\,(vec\,(P)\,, vec\,(P+))$ and $similarity\,(P, P+) = \cos\,(\theta\,(vec\,(P)\,, vec\,(P+)))$. The result by manual is compared with the result by calculation of feature vectors. If $d\,(vec\,(P)\,, vec\,(P+)) < d\,(vec\,(P)\,, vec\,(P-))$ happens in the result by calculation, this shows that $\mathbf{P}^{+}$ is more similar to $\mathbf{P}$ than $\mathbf{P}^{-}$, which



(a) by PCA (b) by T-SNE

FIGURE 13. Dimensionality reduction on CVSM by PCA and T-SNE. Different colors denote different patent clusters. It is obvious to see that results by T-SNE is better than the other one.

TABLE 2. Accuracy rate in term of Euclidean distance and Cosine similarity on manual patent triads.

Accuracy Rate / method	Euclidean Distance	Cosine similarity
CVSM	**94.78%**	93.91%
FVSM	86.96%	84.67%
TF-IDF-based VSM	73.67%	68.33%

means these two results are consistent. In other words, CVSM achieves the goal of patent similarity comparison. The proportion of the number of patent triads with consistent results in the total 300 patent triads is calculated as the accuracy rate of CVSM in the application of patent similarity. The results are shown in Table 2.

As expected, the accuracy rates of the proposed CVSM are consistently higher than those of CNN-based FVSM and TF-IDF-based VSM by the computing method of both Euclidean distance and Cosine similarity. The best experimental result of patent similarity is 94.78%. This indicates that the proposed CVSM can perform better in patent similarity comparison.

D. PATENT CLUSTERING AND NOVEL PATENT MAP GENERATION

A significant application of CVSM is to generate the patent map based on feature vectors. By visualizing patents in a figure, it allows researchers to perform patent analysis intuitively and globally. Patent map as a method of patent visualization has been applied in many works, such as [62]–[64]. Compared with the patent map based on traditional VSMs and FVSM, the patent map based on CVSM can not only retain the original function, i.e., dividing patents into several patent clusters representing different technical fields, but also add new function such as analyzing the relation of citation data. A specific patent and its citation patents are marked on the patent map for analyzing the history of patent evolution and grasping technology trends for reference.

K-means method is used for patent clustering of the patent map. The number of 5 to 50 clusters is tried in our data set. Within-cluster sum of squared errors (SSE) distance is used to measure the performance of K-means results as shown in Fig. 12. Known by the elbow method [65], it can be found the result of 10 clusters has the best performance. Then, keywords in each clusters are extracted for analyzing the technology field of this cluster respectively. The result is shown in Table 3.

Dimensionality reduction algorithms such as the linear dimensionality reduction method like principal component analysis (PCA) [66] and the non-linear dimensionality reduction method like T-distributed stochastic neighbor embedding (T-SNE) [67] are used to reduce the feature vectors of patents to two dimensions and draw the patent map. Since it is obvious to see that the performance of T-SNE (Fig. 13(b)) is better than the one of PCA (Fig. 13(a)), the result of T-SNE is chosen for generating patent map, as shown in Fig. 14.

In the patent map, each patent cluster represents the corresponding technology field of these patents, which is listed in Table. 3. The inferred technology fields of the patent clusters are labeled in the patent map. Besides, each color represents a patent cluster. It can be found that the distance between two similar patent clusters is smaller than that between two irrelevant patent clusters, which meets our expected goals. For example, the cluster of ''network optimization'' and ''network security'' have a close relationship. Thus, they are next to each other.

Moreover, there are several characteristic applications in the patent map for the analysis of patent citation trees, such as tracing the development history of patents. For example, three patents in the patent citation tree of US 9444892 are all related to wireless vehicle communication. However, these patents belong to three clusters, which means different

TABLE 3. Top eight keywords and technology fields of ten patent clusters.

Clusters	1		2		3		4		5	
Keywords	encrypt	access	data	block	packet	switch	pixel	array	terminal	node
	host	key	stream	buffer	node	flow	display	color	remote	connect
	resource	virtual	frame	header	port	route	light	image	module	unit
	client	server	symbol	field	memory	resource	camera	focal	network	access
Technology Fields	network security		data transmission		network optimization		image processing		access control	
Clusters	**6**		**7**		**8**		**9**		**10**	
Keywords	media	video	signal	antenna	mobile	call	user	message	light	wave
	television	event	analog	channel	wireless	post	content	custom	device	surface
	content	multimedia	wavelength	power	remote	portable	device	detect	guid	semiconductor
	social	object	communication	filter	radio	telephone	locate	authenticate	attach	module
Technology Fields	multimedia		radio frequency		mobile communication		application entity		hardware devices	

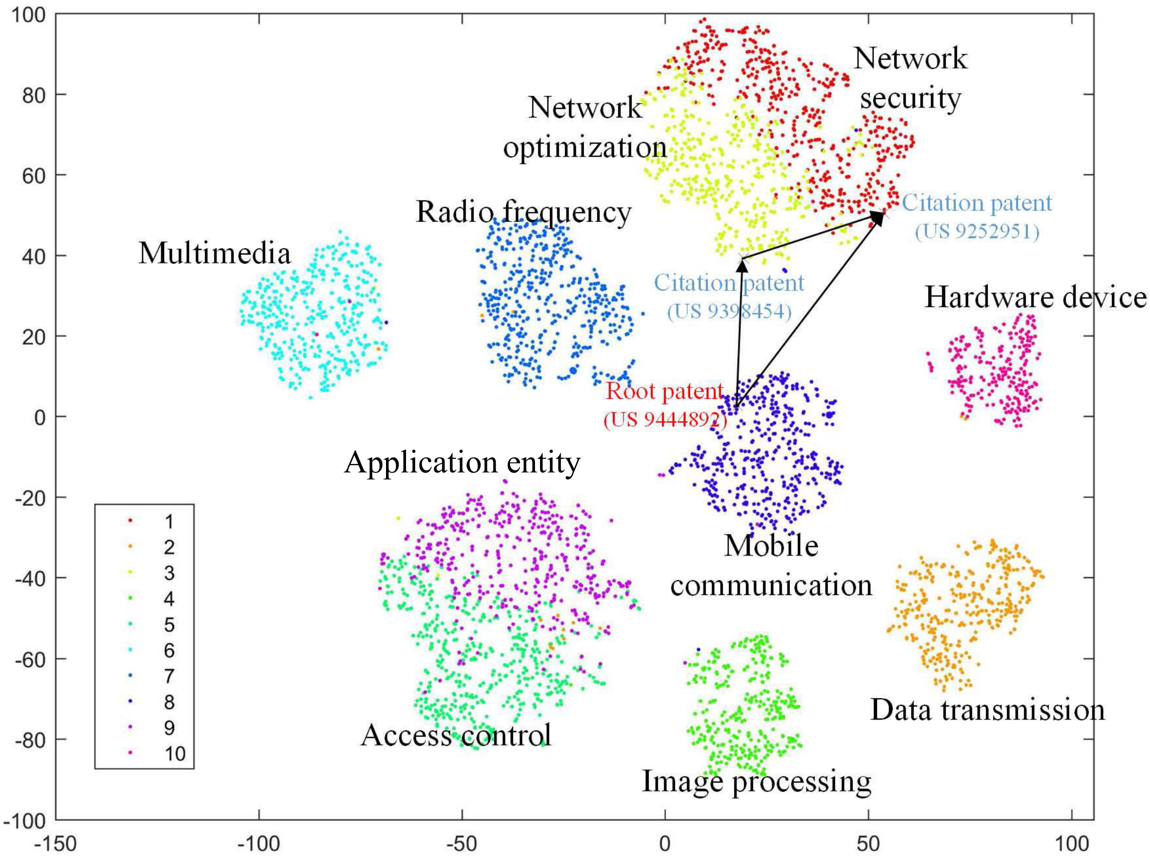


FIGURE 14. The patent map of patents in $\mathbb{S}_B$ as the experiment result.

technology fields. Concretely speaking, the technology of US 9444892 is mainly about mobile communication, while US 9398454 is associated with network optimization, and US 9252951 focuses on network security. The structure of the citation trees can be visualized in the patent map. Even if these three patents vary widely, the route of technological evolution can be displayed intuitively in the patent map. It can be found that the technology of mobile communication in US 9444892 is associated with another two technologies closely, i.e., network optimization and network security. By way of generating the patent map, researchers can intuitively understand the technical development of a patent based on its citation tree.

VI. CONCLUSION

In this paper, we have proposed a new method for citation modeling to make use of both citations and text for patent analysis and established patent citation trees as the modeling result. On this basis, a citation-based convolution neural network namely CCNN is specially designed for mining abstract features in patent citation trees. Several pre-processing methods which convert citation trees to matrices as the input of CCNN have been presented. The citation trees of a subset in USPTO patents have been constructed as the experiment data set. Then, CCNN has been well-trained to retrieve the feature vectors of the data set and used to construct the citation-based VSM, i.e., CVSM. In our experiments, several applications of CVSM, i.e., patent similarity comparison, patent clustering and patent map generation, have been carried out to validate the performances of CVSM. In the case of patent similarity comparison, the accuracy of CVSM is up to 94.78% and superior to that of FVSM. Moreover, the patent map can be conveniently visualized based on the reduced dimensional feature vectors from CVSM.

REFERENCES

[1] A. Abbas, L. Zhang, and S. U. Khan, "A literature review on the state-of-the-art in patent analysis," *World Patent Inf.*, vol. 37, pp. 3–13, Jun. 2014.
[2] P. Moser, "Patents and innovation: Evidence from economic history," *J. Econ. Perspect.*, vol. 27, no. 1, pp. 23–44, Feb. 2013.
[3] S. Nagaoka, K. Motohashi, and A. Goto, "Patent statistics as an innovation indicator," in *Handbook of the Economics of Innovation*, vol. 2. Amsterdam, The Netherlands: Elsevier, 2010, pp. 1083–1127.
[4] G. J. Kim, S. S. Park, and D.-S. Jang, "Technology forecasting using topic-based patent analysis," *J. Sci. Ind. Res.*, vol. 74, pp. 265–270, May 2015.
[5] T. U. Daim, G. Rueda, H. Martin, and P. Gerdsri, "Forecasting emerging technologies: Use of bibliometrics and patent analysis," *Technol. Forecasting Social Change*, vol. 73, no. 8, pp. 981–1012, Oct. 2006.
[6] M. Kim, Y. Park, and J. Yoon, "Generating patent development maps for technology monitoring using semantic patent-topic analysis," *Comput. Ind. Eng.*, vol. 98, pp. 289–299, Aug. 2016.
[7] T. Silva, M. Jian, and Y. Chen, "Process analytics approach for R&D project selection," *ACM Trans. Manage. Inf. Syst. (TMIS)*, vol. 5, no. 4, pp. 1–34, 2014.
[8] A. Shinmori, M. Okumura, Y. Marukawa, and M. Iwayama, "Patent claim processing for readability: Structure analysis and term explanation," in *Proc. ACL- Workshop Patent Corpus Process., Assoc. Comput. Linguistics*, vol. 20, 2003, pp. 56–65.
[9] P. Mahdabi and F. Crestani, "Patent query formulation by synthesizing multiple sources of relevance evidence," *ACM Trans. Inf. Syst. (TOIS)*, vol. 32, no. 4, pp. 1–30, Oct. 2014.
[10] J. Joung and K. Kim, "Monitoring emerging technologies for technology planning using technical keyword based analysis from patent data," *Technol. Forecasting Social Change*, vol. 114, pp. 281–292, Jan. 2017.
[11] J. Hu, S. Li, Y. Yao, L. Yu, G. Yang, and J. Hu, "Patent keyword extraction algorithm based on distributed representation for patent classification," *Entropy*, vol. 20, no. 2, p. 104, Feb. 2018.
[12] Z. Xie and K. Miyazaki, "Evaluating the effectiveness of keyword search strategy for patent identification," *World Patent Inf.*, vol. 35, no. 1, pp. 20–30, Mar. 2013.
[13] S. Nam and K. Kim, "Monitoring newly adopted technologies using keyword based analysis of cited patents," *IEEE Access*, vol. 5, pp. 23086–23091, 2017.
[14] H. Noh, Y. Jo, and S. Lee, "Keyword selection and processing strategy for applying text mining to patent analysis," *Expert Syst. Appl.*, vol. 42, no. 9, pp. 4348–4360, Jun. 2015.
[15] S. Lee, B. Yoon, and Y. Park, "An approach to discovering new technology opportunities: Keyword-based patent map approach," *Technovation*, vol. 29, nos. 6–7, pp. 481–497, Jun. 2009.
[16] D. Eisinger, G. Tsatsaronis, M. Bundschus, U. Wieneke, and M. Schroeder, "Automated patent categorization and guided patent search using IPC as inspired by MeSH and PubMed," *J. Biomed. Semantics*, vol. 4, no. 1, p. S3, 2013.
[17] K. A. Younge and J. M. Kuhn. (2016). *Patent-to-Patent Similarity: A Vector Space Model*. [Online]. Available: https://ssrn.com/abstract=2709238
[18] L. Lei, J. Qi, and K. Zheng, "Patent analytics based on feature vector space model: A case of IoT," *IEEE Access*, vol. 7, pp. 45705–45715, 2019.
[19] S. Brügmann, N. Bouayad-Agha, A. Burga, S. Carrascosa, A. Ciaramella, M. Ciaramella, J. Codina-Filba, E. Escorsa, A. Judea, S. Mille, A. Müller, H. Saggion, P. Ziering, H. Schütze, and L. Wanner, "Towards content-oriented patent document processing: Intelligent patent analysis and summarization," *World Patent Inf.*, vol. 40, pp. 30–42, Mar. 2015.
[20] M. Okamoto, Z. Shan, and R. Orihara, "Applying information extraction for patent structure analysis," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.-SIGIR*, 2017, pp. 989–992.
[21] M. Karki, "Patent citation analysis: A policy analysis tool," *World Patent Inf.*, vol. 19, no. 4, pp. 269–272, Dec. 1997.
[22] X. Li, H. Chen, Z. Huang, and M. C. Roco, "Patent citation network in nanotechnology (1976–2004)," *J. Nanoparticle Res.*, vol. 9, no. 3, pp. 337–352, May 2007.
[23] R. Fontana, A. Nuvolari, and B. Verspagen, "Mapping technological trajectories as patent citation networks. An application to data communication standards," *Econ. Innov. New Technol.*, vol. 18, no. 4, pp. 311–336, Jun. 2009.
[24] Y.-R. Li, L.-H. Wang, and C.-F. Hong, "Extracting the significant-rare keywords for patent analysis," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5200–5204, Apr. 2009.
[25] C. Lee, B. Song, and Y. Park, "How to assess patent infringement risks: A semantic patent claim analysis using dependency relationships," *Technol. Anal. Strategic Manage.*, vol. 25, no. 1, pp. 23–38, Jan. 2013.
[26] D. Korobkin, S. Fomenkov, A. Kravets, S. Kolesnikov, and M. Dykov, "Three-steps methodology for patents prior-art retrieval and structured physical knowledge extracting," in *Creativity in Intelligent Technologies and Data Science* (Communications in Computer and Information Science), vol. 535. Springer, 2015, pp. 124–136.
[27] A. G. Kravets, S. S. Vasiliev, and D. V. Shabanov, "Research of the LDA algorithm results for patents texts processing," in *Proc. 9th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Jul. 2018, pp. 1–6.
[28] D. M. Korobkin, S. A. Fomenkov, and A. G. Kravets, "Methods for extracting the descriptions of sci-tech effects and morphological features of technical systems from patents," in *Proc. 9th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Jul. 2018, pp. 1–4.
[29] J.-L. Wu, P.-C. Chang, C.-C. Tsao, and C.-Y. Fan, "A patent quality analysis and classification system using self-organizing maps with support vector machine," *Appl. Soft Comput.*, vol. 41, pp. 305–316, Apr. 2016.
[30] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*. [Online]. Available: https://arxiv.org/abs/1408.5882
[31] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," 2014, *arXiv:1404.2188*. [Online]. Available: https://arxiv.org/abs/1404.2188
[32] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9745/9552
[33] X. Wang, W. Jiang, and Z. Luo, "Combination of convolutional and recurrent neural network for sentiment analysis of short texts," in *Proc. COLING, 26th Int. Conf. Comput. Linguistics, Tech. Papers*, 2016, pp. 2428–2437.
[34] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975. [Online]. Available: http://doi.acm.org/10.1145/361219.361220
[35] Q. Guo, "The similarity computing of documents based on VSM," in *Proc. Int. Conf. Netw.-Based Inf. Syst.* Berlin, Germany: Springer, 2008, pp. 142–148.
[36] P. D. Turney, "Measuring semantic similarity by latent relational analysis," 2005, *arXiv:cs/0508053*. [Online]. Available: https://arxiv.org/abs/cs/0508053
[37] A. Hliaoutakis, G. Varelas, E. Voutsakis, E. G. Petrakis, and E. Milios, "Information retrieval by semantic similarity," *Int. J. Semantic Web Inf. Syst.*, vol. 2, no. 3, pp. 55–73, Jul. 2006.
[38] H. Uchida, A. Mano, and T. Yukawa, "Patent map generation using concept-based vector space model," in *Proc. NTCIR*, 2004. [Online]. Available: https://pdfs.semanticscholar.org/03d3/729dc9eb4b8a27a62431b2e04da05fcb8b24.pdf
[39] Y.-L. Chen and Y.-T. Chiu, "An IPC-based vector space model for patent retrieval," *Inf. Process. Manage.*, vol. 47, no. 3, pp. 309–322, May 2011.
[40] H. Choi, S. Oh, S. Choi, and J. Yoon, "Innovation topic analysis of technology: The case of augmented reality patents," *IEEE Access*, vol. 6, pp. 16119–16137, 2018.
[41] A. Fujii, "Enhancing patent retrieval by citation analysis," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.-SIGIR*, 2007, pp. 793–794.

[42] P. Érdi, K. Makovi, Z. Somogyvári, K. Strandburg, J. Tobochnik, P. Volf, and L. Zalányi, "Prediction of emerging technologies based on analysis of the US patent citation network," *Scientometrics*, vol. 95, no. 1, pp. 225–242, Apr. 2013.
[43] M. P. Carpenter, F. Narin, and P. Woolf, "Citation rates to technologically important patents," *World Patent Inf.*, vol. 3, no. 4, pp. 160–163, Oct. 1981.
[44] B. Verspagen, "Mapping technological trajectories as patent citation networks: A study on the history of fuel cell research," *Adv. Complex Syst.*, vol. 10, no. 1, pp. 93–115, Mar. 2007.
[45] J. Bakker, "On the relation between patent citations and patent value," in *Proc. KU Leuven, Fac. Econ. Bus., MSI_1611*, 2016. [Online]. Available: http://dx.doi.org/10.2139/ssrn.2874362
[46] I. Von Wartburg, T. Teichert, and K. Rost, "Inventive progress measured by multi-stage patent citation analysis," *Res. Policy*, vol. 34, no. 10, pp. 1591–1607, Dec. 2005.
[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556
[48] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[50] L. Mou, G. Li, L. Zhang, T. Wang, and Z. Jin, "Convolutional neural networks over tree structures for programming language processing," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11775/11735
[51] L. Mou, H. Peng, G. Li, Y. Xu, L. Zhang, and Z. Jin, "Discriminative neural sentence modeling by tree-based convolution," 2015, *arXiv:1504.01106*. [Online]. Available: https://arxiv.org/abs/1504.01106
[52] L. Mou, R. Men, G. Li, Y. Xu, L. Zhang, R. Yan, and Z. Jin, "Natural language inference by tree-based convolution and heuristic matching," 2015, *arXiv:1512.08422*. [Online]. Available: https://arxiv.org/abs/1512.08422
[53] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: https://arxiv.org/abs/1409.0473
[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[55] USPTO. (2019). *Bulk Data Storage System (bdss) Version 1.1.0*. [Online]. Available: https://bulkdata.uspto.gov/
[56] E. Pouget, G. Pibre, and S. Marrot, "Photoactivatable system for inhibiting hydrosilylation," U.S. Patent 9 834 648, Dec. 5, 2017.
[57] N. H. Ali and N. S. Ibrahim, "Porter stemming algorithm for semantic checking," in *Proc. 16th Int. Conf. Comput. Inf. Technol.*, 2012, pp. 253–258.
[58] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[59] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction," in *Proc. Int. Work-Conf. Artif. Neural Netw. (IWANN)*. Barcelona, Spain: Springer, 2005, pp. 758–770.
[60] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. Ijcai*, 1995, vol. 14, no. 2, pp. 1137–1145.
[61] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, "Using of Jaccard coefficient for keywords similarity," in *Proc. Int. Multiconf. Eng. Comput. Scientists*, 2013, vol. 1, no. 6, pp. 380–384.
[62] R. Chen, "Design patent map visualization display," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 12362–12374, Dec. 2009.
[63] C. Lee, B. Kang, and J. Shin, "Novelty-focused patent mapping for technology opportunity analysis," *Technol. Forecasting Social Change*, vol. 90, pp. 355–365, Jan. 2015.
[64] P. F. Huang, C. G. Wei, and R. N. Pan, "Patent map analysis of measurement technology," *Key Eng. Mater.*, vol. 680, pp. 60–67, Feb. 2016.
[65] P. E. Green, J. Kim, and F. J. Carmone, "A preliminary study of optimal variable weighting in k-means clustering," *J. Classification*, vol. 7, no. 2, pp. 271–285, Sep. 1990.
[66] I. Jolliffe, *Principal Component Analysis*. Berlin, Germany: Springer, 2011, pp. 1094–1096. [Online]. Available: https://doi.org/10.1007/978-3-642-04898-2_455
[67] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

JIAJU QI received the B.S. degree from the Beijing University of Posts and Telecommunications (BUPT), China, in 2017. He is currently pursuing the Ph.D. degree with the Intelligent Computing and Communication Lab, Key Laboratory of Universal Wireless Communications, Ministry of Education, BUPT. His research interests include machine learning in the Internet of Things (IoT) networks and patent analysis.

LEI LEI (Senior Member, IEEE) received the B.S. and Ph.D. degrees in telecommunications engineering from the Beijing University of Posts and Telecommunications, China, in 2001 and 2006, respectively. She is currently with the University of Guelph, Canada. Her current research interests include machine learning, the Internet of Things, and mobile cloud computing.

KAN ZHENG (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Beijing University of Posts and Telecommunications (BUPT), China, in 1996, 2000, and 2005, respectively. He is currently a Professor with BUPT. He has rich experiences on the research and standardization of the new emerging technologies. He is the author of more than 200 journal articles and conference papers in the field of the Internet of Things (IoT), vehicular communication, and security. He holds editorial board positions for several journals and has organized several special issues.

XIANBIN WANG (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from the National University of Singapore, in 2001.

He is currently a Professor and the Tier-I Canada Research Chair with Western University, Canada. Prior to joining Western, he was with the Communications Research Centre Canada (CRC), as a Research Scientist/Senior Research Scientist, from July 2002 and December 2007. From January 2001 to July 2002, he was a System Designer with STMicroelectronics, where he was responsible for the system design of DSL and Gigabit Ethernet chipsets. His current research interests include 5G technologies, the Internet of Things, communications security, machine learning, and locationing technologies. He has over 300 peer-reviewed journal and conference papers and holds 26 granted and pending patents and several standard contributions. He is a Fellow of Canadian Academy of Engineering and an IEEE Distinguished Lecturer. He has received many awards and recognitions, including the Canada Research Chair, the CRC President's Excellence Award, the Canadian Federal Government Public Service Award, the Ontario Early Researcher Award, and five IEEE best paper awards. He was involved in many IEEE conferences, including GLOBECOM, ICC, VTC, PIMRC, WCNC, and CWIT, in different roles, such as the Symposium Chair, a Tutorial Instructor, the Track Chair, the Session Chair, and the TPC Co-Chair. He currently serves as an Editor/Associate Editor of the IEEE Transactions on Communications, the IEEE Transactions on Broadcasting, and the IEEE Transactions on Vehicular Technology. He was also an Associate Editor of the IEEE Transactions on Wireless Communications, from 2007 to 2011, and the IEEE Wireless Communications Letters, from 2011 to 2016.