IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 25, 2019, accepted March 24, 2019, date of publication March 27, 2019, date of current version April 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2907815

Unstructured Text Resource Access Control Attribute Mining Technology Based on Convolutional Neural Network

AODI LIU, XUEHUI DU, AND NA WANG
[1]National Digital Switching System Engineering and Technological Research Center, Zhengzhou 450000, China
[2]Zhengzhou Science and Technology Institute, Zhengzhou 450000, China

Corresponding author: Xuehui Du (dxh37139@sina.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0803603 and Grant 2016YFB0501901, in part by the Natural Science Foundation of Henan Province of China under Grant 162300410334, and in part by the National Natural Science Foundation of China under Grant 61802436.

ABSTRACT In the attribute-based access control (ABAC) model, attributes are the basis for controlling access to data resources. The existing attribute extraction methods that are based on manual management are time consuming and have a high cost, variable accuracy, and poor scalability when dealing with massive unstructured text from big data resources. This paper proposes a multidimensional hybrid feature generation method for text resource attributes. The method comprehensively calculates the characteristics of attributes themselves, the relationships between attributes, and the relationship between attributes and resources. It can fully and accurately characterize the attributes. It converts attribute features into grayscale images in order to translate attribute mining problems into image recognition problems. We propose an attribute mining method based on a convolutional neural network (CNN). We use neural networks to automatically correlate features. It means there is no need to manually consider the importance of features and their relationships. This avoids the need for security experts to manually label the attributes of massive resources and facilitates the automatic and intelligent mining of ABAC resource attributes. The experimental results show that compared with the benchmark algorithm, the proposed method has improved accuracy and recall rate and can provide attribute support for ABAC of big data resources.

INDEX TERMS Attribute-based access control (ABAC), attribute mining, deep learning, unstructured data.

I. INTRODUCTION

Recently the rapid development of big data technology has received more and more attention. While big data can provide convenience for people, for example, by suggesting things they may like, it also brings huge security and privacy risks. How to effectively protect the security of big data resources has become a challenge for big data promotion and its application. Access control technology [1] is one of the core technologies for providing big data security. Access controls prevent the unauthorized use of data resources by managing user permissions. Attribute-Based Access Control (ABAC) [2] uses the attributes of the subject, resource, operation and environment as the basic elements of access control. It can flexibly utilize the set of attributes owned by the subject and or the resource to determine whether to grant access or not. Attributes can strongly express semantics. ABAC can better separate policy management from the access control decision and is compatible with traditional mechanisms such as discretionary, mandatory and role based access control (DAC, MAC, and RBAC) [3], [4]. It is suitable for solving fine-grained access control and large-scale dynamic authorization in big data computing environments.

Attributes are the foundation and core of the ABAC mechanism. Attributes are divided into subject attributes, resource attributes, operational attributes, and environmental attributes. The number of subject attributes, action attributes, and environment attributes is relatively limited. But the number of resource attributes used to describe massive data resources can be enormous. Unlike data resources in traditional information systems, big data resources are composed of structured, semi-structured and unstructured

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Yu.

multivariate data. As an important part of the big data set, unstructured data is exploding at an alarming rate. According to the IDC survey [5], 80% of the data in the enterprise is unstructured data, which is growing at an exponential rate of 60 percent a year. Especially in social big data, the Internet of Things big data, medical big data, and industrial big data, unstructured data often exhibits features such as: real-time, dynamic, multi-source, and heterogeneous. These features provide new requirements for the automatic, scalable, and adaptable application of ABAC technologies [6]. In order to implement efficient fine-grained ABAC for big data resources, it is necessary to conduct further research into the attribute extraction methods of unstructured data resource objects. How to achieve automatic attribute extraction from unstructured big data resource objects has become a key problem that needs to be solved urgently.

Existing research into the attribute extraction from unstructured big data has identified the following two main challenges.

(1) It is difficult to accurately describe the unstructured resource objects being access controlled, which challenges the implementation of fine-grained access control. A premise of ABAC is that the resource's attributes can be identified and managed. When controlling access to structured data, the resource object is often described by its data type attribute, for example, user A cannot access resources of data type B. This is convenient and feasible for pre-processed structured data. Unfortunately, unstructured data only carries basic attributes such as generation time and generator, and lacks business attributes describing the internal characteristics and types of the resources. Research is needed into methods for extracting a resource's attributes.

(2) It is difficult for security experts to manually identify and manage the attributes of massive and dynamic data resources, which poses a challenge to the implementation of dynamic access control. Existing attribute management technologies mainly rely on the experience of security experts to manually annotate and manage resource attributes. However, in the complex big data computing environment [7], the scale of object resources is huge, and the growth rate is extremely fast, making it even more difficult for security experts to implement such labor-intensive attribute marking and management schemes [8], [9]. In addition, the quality of attribute selection directly affects the performance of ABAC. Therefore, it is urgent to: relieve security personnel from this onerous attribute management work, reduce the dependence of attribute management on the professional knowledge of security experts, and improve the reliability and accuracy of attribute management.

Since the unstructured data has endogenous attributes, it can depict the characteristics of the resource object itself. For example, in social big data [10], [11], unstructured ''message data'' published by different types of users at different times tends to be different in its attributes; in medical big data [12], [13], the attributes of unstructured ''case data'' for patients with different types of diseases are also fundamentally different. Since text data is a typical unstructured data resource, this paper mainly solves the problem of extracting text resource attributes. Automatically mining the attributes of unstructured text data realizes the objective of accurately describing text resources. Configuring these attributes into the predicates of access control rules, results in accurate and efficient dynamic fine-grained access control for big data resources. Consequently, this paper studies the problem of unstructured text big data attribute extraction. Our contribution includes:

- the design of a feature generation method based on multi-dimensional hybrid features, and
- a neural network-based access control attribute mining mechanism, called AM_NET.

The attributes in the unstructured text are converted into grayscale images, and the attribute mining problem is transformed into the two-class/binary classification problem. A convolutional neural network (CNN) is used to further extract the features and relationships of the grayscale images, and realize the automatic mining of resource attributes. The experimental results show that the proposed method can achieve an accuracy of 90.57%, and it has certain advantages in precision, recall and F1 compared with the benchmark methods. It can effectively support intelligent decision support for management of attributes for ABAC in big data environments.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 formalizes the key issues that are solved by attribute mining and it introduces the implementation framework in detail. Section 4 proposes the multi-dimensional mixed feature generation method. Section 5 proposes the attribute mining mechanism based on a CNN. Section 6 presents the experiments and related analysis of the proposed method. Finally, in section 7 we summarize our research.

II. RELATED WORK

In the field of access control attribute mining, there are many papers on subject role mining [14], [15], but the research on resource attribute mining is minimal. Using the combined keywords of ''access control'' and ''attribute mining'', no published literature was directly retrieved from databases such as Google Scholar, Web of Science, and CNKI. However, similar publications do exist in the fields of knowledge reasoning, network data attribute discovery, and natural language processing.

In the field of knowledge reasoning, Lee *et al.* [16] proposed a concept-based attribute mining method. This method supports knowledge reasoning by using Bayesian probabilities to realize the automatic acquisition of attributes from different knowledge concepts, and evaluates the importance of corresponding attributes. Li *et al.* [17] proposed an attribute mining method based on unsupervised compact clustering for mining entity attribute synonyms, which achieved optimal attribute acquisition. In the field of network data attribute discovery, Ming [18] proposed a latent attribute information

discovery method based on link semantics for vertical search website data. Guo-Qing and Jian-Hua [19] used pre-defined rules and evidence to achieve the discovery of new attributes of Web pages based on credibility analysis. In the field of natural language processing, text keywords or topics are important attributes describing text resources. Text keyword extraction [20], [21] automatically extracts words or phrases that are related to the topic or are of great importance from the collected text. Keyword extraction is basic and necessary work in the field of natural language processing, and its research results are widely used in data retrieval, abstract generation, topic analysis and so on. Commonly used text keyword extraction methods mainly include unsupervised methods and supervised methods.

Unsupervised methods include statistical-based methods (such as Term Frequency-Inverse Document Frequency (TF-IDF) [22]–see section 4.1 for more details), graph structure-based methods (such as TextRank [23]), topic-based methods (such as Latent Dirichlet Allocation (LDA) [24]), and their improved methods. Huang *et al.* [25] introduced the intra-distribution degree DI (Distribution Information) to adjust the feature weight of the vocabulary, and improved the traditional TF-IDF extraction method. Liu and Peng [26] improved the performance of TF-IDF by adding emotional judgments of positive and negative examples. Ma *et al.* [27] transformed the text into a word graph structure, and proposed an extraction algorithm based on random walk. The node weights were calculated according to the node correlation degree and co-occurrence distance in the graph, which is better in short text experiments. Florescu and Caragea [28] integrated the positional information of words into a biased PageRank algorithm. To solve the problem of lacking global computation in existing methods, Boudin [29], [30] converts the keyword extraction into a combinatorial optimization problem, which is solved by using an Integer Linear Program (ILP). The method only uses the average value of TF-IDF, TextRank and Logistic Regression as the weight value, so its scalability is poor. Saeidi *et al.* [31] introduced uncertainty to improve the performance of LDA methods for input data in the presence of noise and distortion. In the field of supervised extraction, the key words are extracted into machine learning problems, and the model is trained based on a manually labeled corpus. Haddoud and Said [32] introduced a document maximum index (DPM-index) to develop a supervised learning system using 18 statistical features. In the keyword extraction of scientific literature, Caragea *et al.* [33] extracted 9 features by combining literature citations with statistical features, and implemented keyword extraction by a Naive Bayesian method. Gollapalli *et al.* [34] incorporated expert knowledge into feature selection and implemented supervised keyword extraction based on CRFs (conditional random field). Meng *et al.* [35] proposed a keyphrase prediction generation model based on the encoder-decoder framework. It can overcome the lack of semantic information in other methods, and can generate missing keyphrases according to the text. Other supervised extraction methods include decision trees [36], random forests [37], Support Vector Machine [38], and so on.

The unsupervised method is simple and easy to implement. However, only the word structure in a single text is considered. The semantic information inherent in the text cannot be reflected. The extraction effect is poor compared to the supervised method. However, most existing supervised methods take a single feature with a practical meaning, or a limited number of features (the number of features is generally less than 20) for attribute extraction. There are some shortcomings such as less feature quantity, poor robustness and weak expansion. There is a lack of consideration of other multi-dimensional features in the text. The attribute description is not accurate enough, and the actual effect needs to be improved.

Based on the above analysis, the current research on attribute mining in the field of text big data access control is still in its infancy. Existing text keyword extraction technology is of interest to our research. However, the direct application of text keyword extraction technology for text attribute mining has problems due to the low accuracy of attribute mining, the large computational cost of attribute mining, and the difficulty in mining dynamic resource attributes.

III. PROBLEM DEFINITION AND IMPLEMENTATION FRAMEWORK

A. PROBLEM DEFINITION

The access control attribute mining problem is defined as follows:

Let $R = \{r_1, r_2, \ldots, r_n\}$ be a set of n text resources that need to be protected by access control. Each resource $r_i \in R$ has a set of m attributes $A = \{a_{i,1}, a_{i,2}, \ldots, a_{i,m}\}$, and the target of attribute mining is:

(1) Generate candidate attribute sets $C_i = \{c_{i,1}, c_{i,2}, \ldots, c_{i,m}\}$.

(2) Find a function that maps the candidate attribute $c_{i,j} \in C_i$ to a category or score. Then, the most representative key attribute set in the resource r_i is extracted from the candidate attribute set according to the category or the score.

B. IMPLEMENTATION FRAMEWORK

The implementation framework for attribute mining is shown in Figure 1. The specific process is as follows:

(1) Text resource preprocessing. Preprocessing includes: calculation of word segmentation and part-of-speech tagging, dependency analysis, named entity recognition for the text resource training set, removal of redundant and invalid words, and analysis of the attribute semantic associations within the text. This step can significantly affect the accuracy of the attribute mining method.

(2) Generate a candidate attribute set. Based on the attribute semantic association analysis in the text, stop words and words of a particular part-of-speech are removed, and the

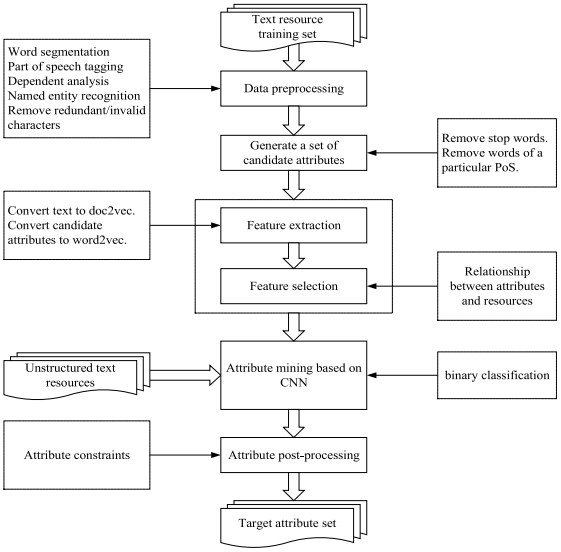


FIGURE 1. Access control attribute mining framework based on mixed features.

candidate attribute set of top 20 is extracted based on the TF-IDF algorithm.

(3) Feature extraction and selection. The Skip-gram neural network is used to train the language model. Training of language model mainly includes two kinds of models: CBOW (Continuous bag-of-Words) and Skip-Gram. In the CBOW, surrounding words are used to predict the center word. In Skip-gram, the central word is used to predict the words around it. The Skip-gram makes more predictions. However, as in skip-gram, each word is affected by the words around it. Each word as the central word, will carry out many times of prediction and adjustment. Therefore, this multiple adjustment will make the word vector relatively more accurate. All the attributes and texts in the training set are mapped to the abstract high-dimensional word vector space and text vector space. The multi-dimensional attribute features, such as attributes and relationships between attributes, and relationships between attributes and resources, are computed and extracted. This produces the attribute feature vector.

(4) Attribute mining engine. The attribute mining problem is transformed into a two-class/binary classification problem of whether the candidate attribute is a key attribute or not. We convert the feature vector of the candidate attributes into a representation of the grayscale image. The classifier is trained using a CNN to obtain a trained attribute mining model.

(5) Model evaluation and application. After evaluating the performance of the attribute mining engine through a test set, it is applied to the attribute mining processing of real big data resources.

IV. MULTI-DIMENSIONAL HYBRID FEATURE GENERATION METHOD

We designed a multi-dimensional hybrid feature generation method to extract as many text-related features as possible, so as to more accurately characterize the attributes of unstructured text resources. The hybrid features can be divided into: general features, structural features, and semantic features, which are used to represent attributes, relationships among attributes, and relationships between attributes and resources. These three categories of features include a total of 40 specific feature categories. A 100-dimensional attribute feature vector is constructed to characterize the candidate attributes.

A. GENERAL FEATURES

Through the analysis of text attributes, combined with the current research results in the field of keyword extraction [39], we selected the following 18 important features as our general features.

A single feature is not sufficient to accurately identify the key attributes. Attributes need to be portrayed from multiple dimensions. For example, the frequency of a key attribute is usually higher than that of a normal attribute, because key attributes often appear more frequently throughout the text. However, the attributes frequently appearing in all texts have high frequency, but they are not special and are not representative of specific texts. The following is an introduction to some of the key general features.

(1) Term Frequency-Inverse Document Frequency (TF-IDF). Where TF (terms frequency) indicates the frequency at which a certain keyword appears, and IDF (inverse document frequency) indicates the level at which the attribute distinguishes different types of document. The TF-IDF value evaluates the importance of an attribute, by calculating the frequency with which the attribute appears in the current document and the frequency with which the attribute appears in different documents.

(2) TextRank. TextRank draws on the PageRank idea and calculates the importance of attributes based on random walks. The candidate attributes are constructed into the nodes of a directed graph. When two attributes appear simultaneously in a co-occurrence window of length K, there is a co-occurrence relationship between the two attributes. Edges are used to connect related attributes, propagate the attribute weights iteratively and get the TextRank value after convergence.

(3) Terms co-occurrence matrix probability skewness

When two attributes appear in a sentence at the same time, we think that the related attributes are co-occurring once. The co-occurrence frequency of the candidate attributes will constitute a diagonal matrix. In order to evaluate the skewness of the co-occurrence probability distribution, a skewness concept is introduced to quantify the direction and extent of the attribute distribution skew.

B. STRUCTURAL FEATURES

Structural features are used to indicate the representativeness of attributes. The foundation of the scheme lies in the structural difference between key attributes and non-key attributes. It is necessary to design an attribute structure representation method to characterize the structural features of the

TABLE 1. General features.

Number	Feature Name	Calculation formula
G1	Text length	R_Len(r)=$\|r\|$， $\|r\|$ is the number of terms contained in resource r.
G2	TF	TF(a,r)=f(a,r)/$\|r\|$， f(a,r) is the number of times the candidate attribute a appears in the resource r
G3	TF ratio	$TF(a,r)=\frac{TF(a,r)}{\max\limits_{c\in comp(a)} TF(c,r)}$， comp($a$) is the set of terms that make up a.
G4	Term length	A_Len(a)=$\|a\|$， $\|a\|$ is the number of words in the candidate attribute.
G5	HF	HF(a,r/4)= f(a,r/4)/$\|r$/4$\|$， r/4 is the first quarter of r， $\|r$/4$\|$ is the number of terms contained in r/4.
G6	HF ratio	$HF(a,r)=\frac{HF(a,r/4)}{\max\limits_{c\in comp(a)} HF(c,r/4)}$
G7	IDF	IDF(a,C)=log($\|C\|$/df(a,C))， df(a,C) is the number of texts containing a in corpus C.
G8	TF-IDF	TF_IDF(a,r,C)=TF(a,r)×IDF(a,C)
G9	TF-IDF ratio	$TF\_IDF\_Ratio(a,r,C)=\frac{TF\_IDF(a,r,C)}{\max\limits_{c\in comp(a)} TF\_IDF(c,r,C)}$
G10	TextRank value	$S(v_i)=(1-d)+d\times\sum\limits_{v_j\in Adj(v_i)}\frac{w_{ji}}{\sum_{k\in Adj(v_j)}w_{jk}}\cdot s(v_j)$， w_{ji} is the weight of the attribute node, d is the damping coefficient.
G11	Number of candidate attributes,	AttrNum(r)=k_a， k_a is the number of candidate attributes included in resource r.
G12	Nmber of sample sentences	SenNum(r)=k_s， k_s is the number of sentences contained in resource r.
G13	Text category	Class(r)=Kmeans(r), Calculate the cluster category of resource r.
G14	Title length	Title_len(title)=$\|title_r\|$， $\|title_r\|$ is the number of terms in the title of resource r.
G15	Co-occurrence skewness	$SK_i=\frac{(k_a-1)\sum_j(m_{i,j}-\bar{m}_i)^3}{(k_a-2)(k_a-3)\cdot SD_i^3}(k_a\geq 4)$， $m_{i,j}$ is the co-occurrence frequency of the attributes i and j, $\bar{m}_i$ is the average value, and SD_i is the standard deviation.
G16	Co-occurrence kurtosis	$Kurt_i=\frac{\frac{1}{k_a}\sum_j(m_{i,j}-\bar{m}_i)^4}{\left(\frac{1}{k_a}\sum_j(m_{i,j}-\bar{m}_i)^2\right)^2}-3$
G17	Co-occurrence difference	$Diff_i=diff(M)$， M is the co-occurrence frequency matrix.
G18	Co-occurrence variance	$Var_i=\frac{\sum_j(m_{i,j}-\bar{m}_i)^2}{k_a}$

candidate attributes. Structural features are complex and difficult to visually analyze. If there are certain differences in the position of an attribute in the document, the location information of the attribute can be used to describe the distribution, span and position of the candidate attribute in the target resource. In general, attributes often appear in specific positions in the text. For example, words that appear at the head of a document and the head of a paragraph have a stronger representation than words that appear elsewhere in the text. Therefore, we introduce the location feature of an attribute, to say whether the attribute appears in the title, whether it appears in the first sentence of the body, whether it appears in the last sentence of the body, and whether it appears in the middle of the text. In addition, we also introduce the frequency of attributes appearing in the first half of the text and the statistical characteristics of the sentence in which the attribute is located. We have selected the following 12 important features as our structural features.

C. LINGUISTIC FEATURES

Linguistic features refer to the features extracted from the morphology (such as the part of speech) and sentence syntax of the candidate attributes. They are obtained through lexical and syntactic analysis for automatic word segmentation, part-of-speech tagging, dependency analysis, and named entity recognition. The details of lexical and syntactic analysis are not the focus of this paper, so we won't go into too much detail about them. We have selected the following 10 important features as linguistic features.

TABLE 2. Structural features.

Number	Feature Name	Calculation formula
S1	Occurrence in title	$s_1 = has\_boolean(tit, a)$, tit is the text title.
S2	Occurrence in first third	$s_2 = has\_boolean(r_{1/3}, a)$, $r_{1/3}$ is the first third of text.
S3	Occurrence in middle third	$s_3 = has\_boolean(r_{2/3}, a)$, $r_{2/3}$ is the middle third of text.
S4	Occurrence in last third	$s_4 = has\_boolean(r_{3/3}, a)$, $r_{3/3}$ is the last third of text.
S5	Special format 1	$s_5 = has\_boolean(a, num)$, *num* represents the number.
S6	Special format 2	$s_6 = has\_boolean(a, abbr)$, *abbr* represents the abbreviations.
S7	Special format 3	$s_7 = has\_boolean(a, guil)$, *guil* represents the guillemet.
S8	Initial position	$FP(a,r) = \frac{1}{\|r\|} \cdot pos(a,r)$, $pos(a,r)$ is the first occurrence position of attribute a in text r.
S9	Average sentence length	$AL(a,r) = \frac{1}{\|S(a,r)\|} \cdot \sum_{s \in S(a,r)} \|s\|$, $S(a,r)$ is the set of all sentences that contain a in resource r, $\|S(a,r)\|$ is the total number of sentences, $\|s\|$ is the length of the sentence
S10	Longest sentence length	$MXP(a,r) = \max_{s \in S(a,r)} (\|s\|)$
S11	Shortest sentence length	$MNL(a,r) = \min_{s \in S(a,r)} (\|s\|)$
S12	Average position of sentence	$SP(a,r) = \frac{1}{\|S(a,r)\|} \cdot \sum_{s \in S(a,r)} pos(s,r)$

TABLE 3. Linguistic features.

Number	Feature Name	Calculation formula
L1	Part-of-Speech	$L_1 = POS\_Tag(a)$, Mark a correct part of speech for each word in the participle result
L2	Proper noun	$L_2 = is\_properNoun(a)$, Determines whether a candidate attribute is a proper noun.
L3	Number of possible POS	$L_3 = POS\_Num(a)$, Calculate the number of possible parts of speech for candidate attributes.
L4	Probability of POS	$L_4 = Pos\_\max\_prob(a)$, Calculate the probability of the part of speech of the candidate attribute.
L5	Named Entity	$L_5 = NER(a)$, Identify named entities for candidate attributes
L6	Cosine similarity of attribute and text	$Sim(W,D) = \frac{\sum_{i=1}^{n}(w\_vec_i \cdot d\_vec_i)}{\sqrt{\sum_{i=1}^{n}(w\_vec_i)^2} \cdot \sqrt{\sum_{i=1}^{n}(d\_vec_i)^2}}$
L7	Euclidean distance of attribute and text	$dist(W,D) = \sqrt{\sum_{i=1}^{n}(w\_vec_i - d\_vec_i)^2}$
L8	Average similarity of attribute	$Similarity(vec) = \frac{1}{m} \cdot \sum_{j=1}^{m} \frac{\sum_{i=1}^{n}(vec \cdot w\_vec_{i,j})}{\sqrt{\sum_{i=1}^{n}(vec)^2} \cdot \sqrt{\sum_{i=1}^{n}(w\_vec_{i,j})^2}}$
L9	Kurtosis of attribute similarity	$Kur = \frac{\sum_{i=1}^{n}(s_i - \bar{s})^4}{(n-1)(SD^4 - 3)}$, s_i is the cosine similarity of attribute i.
L10	Difference of attribute similarity	$Var_i = \frac{\sum_j (s_{i,j} - \bar{s}_i)^2}{k_a}$, k_a is the number of candidate attributes included in resource r.

Semantic similarity calculation is based on the word vector model, the attribute vector Word2vec and the text vector Doc2vec are trained to calculate the relationship between attributes and resources. (w_vec_1, w_vec_2, …, w_vec_n) represents the attribute vector W, (d_vec_1, d_vec_2, …, d_vec_n) represents the text vector D, w_vec_i represents the item of

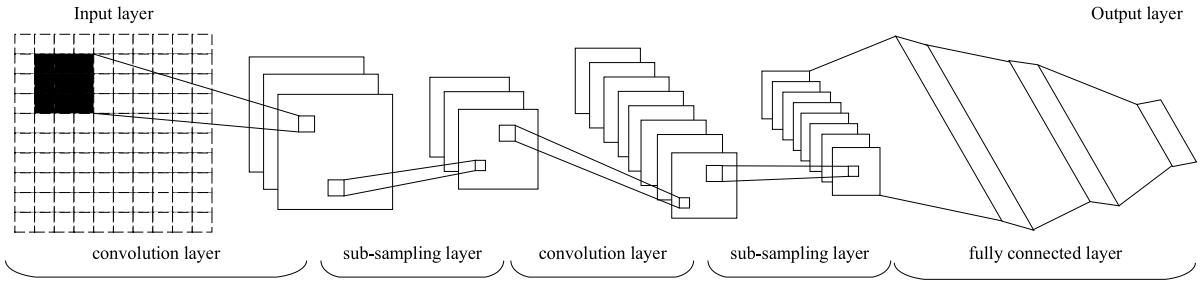


FIGURE 2. Attribute mining CNN for AM-NET.

the vector W, and d_vec_i represents the item of the vector D. Use W and D to compute the feature L6-L10.The cosine similarity (L6 in the table) between the attribute vector and the text vector is the measure of the difference between the two individuals using the cosine of the angle between the two vectors in the vector space.

The Euclidean distance (L7 in the table) is the distance between two points in the n-dimensional Euclidean space. In the language model obtained through neural network word vector training, the attribute vector Word2vec and the text vector Doc2vec contain potential semantic information. Generally speaking, the higher the semantic similarity and the closer the distance.

V. MINING ALGORITHM BASED ON CNN

CNNs [40], [41] are widely used in popular fields such as image recognition, video analysis, and autopilots because of their good learning performance. CNNs have developed rapidly in recent years. A CNN uses spatial local perception and a weight sharing network structure to reduce the complexity and number of parameters in neural network model training. CNNs are more advantageous when the input data feature dimension is high. In the work of this paper, we transform the 100-dimensional attribute feature vector into 10*10 grayscale image as the input to the network, and transform the attribute mining problem into image recognition problem. Compared with traditional algorithms, this method is not limited by feature relationship calculation and data reconstruction, and there is no need to artificially consider the importance of features and their internal relationships. Automatic correlation analysis of features by neural networks is more robust and scalable.

The network structure of AM_NET for grayscale image recognition consists of the components shown in Figure 2. The first component is the input layer, which introduces training images into the neural network. Next are the convolutional layer and the sub-sampling layer. Convolutional layers enhance signal characteristics and reduce data noise. The subsampling layer can reduce the amount of data processing while retaining useful information. Two fully connected layers are then connected together, which converts the two-dimensional features into one-dimensional features that conform to the classifier criteria. Finally, the classifier classifies the candidate attributes according to the characteristics of the candidate attribute image to determine whether it is a key attribute.

A. CONVOLUTION LAYER

A convolutional layer can effectively reduce the number of image parameters while preserving the main features of the image. It can effectively avoid over-fitting and improve the generalization ability of the model. The input is multiple mappings and the output is a dimensionally reduced mapping. Each mapping is a combination of input mapping convolution values belonging to the upper layer and can be given by the following equation.

$$a_j^{(l)} = f(u_j^{(l)}) = f(\sum_{i \in N_j} a_j^{(l-1)} * K_{i,j}^{(l)} + b_j^{(l)}) \tag{1}$$

N_j is the set of input maps, $K_{i,j}^{(l)}$ is the convolution kernel for connecting the i-th input feature map and the j-th output feature map, $b_j^{(l)}$ is the offset term of the j-th feature map, and f is the activation function. The backpropagation error $\delta_j^{(l)}$ is calculated as follows:

$$\delta_j^{(l)} = \beta_j^{(l+1)}(up(\delta_j^{(l+1)}) \circ f'(u_j^{(l)})) \tag{2}$$

The δ of each neuron of the convolutional layer l is only related to the related neurons of the $l+1$ layer, and the convolutional layer l to the pooled layer $l+1$ is subjected to a downsampling operation to reduce the matrix dimension. Therefore, $\delta_j^{(l)}$ needs to adopt $up()$ as the matrix dimension of layer l. $\beta_j^{(l+1)}$ is the sampling weight. The partial derivative of the error cost function for the deviation b and the convolution kernel K is as follows:

$$\frac{\partial E}{\partial b_j} = \sum_{s,t} (\delta_j^{(l)})_{s,t} \tag{3}$$

$$\frac{\partial E}{\partial K_{i,j}^{l}} = \sum_{s,t} (\delta_j^{(l)})_{s,t} (P_i^{(l-1)})_{s,t} \tag{4}$$

$(*)_{s,t}$ is a traversal of all elements of $*$, $(P_i^{(l-1)})_{s,t}$ is a matrix of $a_j^{(l-1)}$ and $K_{i,j}^{(l)}$ convolutional computing elements of l-1 connected by $(\delta_j^{(l)})$. (s, t) is the positional information

of the matrix which can be obtained by calculating the convolution value of the input feature map in the region (s, t) and the convolution kernel $K_{i,j}^{(l)}$.

B. DOWNSAMPLING LAYER

The downsampling layer is also referred to as the pooling layer. Generally, it takes the maximum or average value in the pooled area (called maximum pooling or average pooling respectively). It is not affected by backpropagation. This layer can reduce the influence of image deformation, reduce the dimension of feature mapping, improve the accuracy of the model, and avoid over-fitting. The output of the downsampling layer is as follows:

$$a_j^{(l)} = f(\beta_j^{(l)} down(a_j^{(l-1)} + b_j^{(l)})) \tag{5}$$

where $down(\cdot)$ is the downsampling function and b_j^l is the bias term. The error of backpropagation is calculated as follows:

$$\delta_j^{(l)} = \delta_j^{(l+1)} K_{j,i}^{(l+1)} \circ f'(u^{(l)}) \tag{6}$$

The partial derivative of the error cost function for the bias b and the weight .. can be expressed as follows, where $d_j^{(l-1)} = down(a_j^{(l-1)})$.

$$\frac{\partial E}{\partial b_j^{(l)}} = \sum_{s,t} (\delta_j^{(l)})_{s,t} \tag{7}$$

$$\frac{\partial E}{\partial \beta_j^{(l)}} = \sum_{s,t} (\delta_j^{(l)} \circ d_j^{(l-1)})_{s,t} \tag{8}$$

C. FULLY CONNECTED LAYER

The calculation of the fully connected layer is consistent with the calculation of the common neural network. Its output is as follows:

$$a^{(l+1)} = f(w^{(l+1)} a^{(l)} + b^{(l+1)}) \tag{9}$$

The error of backpropagation is calculated as follows:

$$\delta^{(l)} = w^{(l)} a^{(l)} \cdot \delta^{(l+1)} \tag{10}$$

The partial derivative of the error cost function for the bias b and the weight w is calculated as follows:

$$\frac{\partial E}{\partial b^{(l)}} = \delta^{(l)} \tag{11}$$

$$\frac{\partial E}{\partial w^{(l)}} = \delta^{(l)} (a^{(l-1)})^T \tag{12}$$

In addition, before inputting the attribute grayscale data into the CNN network, the data needs to be normalized to improve the efficiency and accuracy of the model training. The normalization calculation method is as follows:

$$a_{scale} = \frac{a - a_{\min}}{a_{\max} - a_{\min}} \tag{13}$$

where a is the raw data for each feature in the data set, and a_{max} and a_{min} are the maximum and minimum values of the original data set respectively. After the above calculation, we can get the CNN weight update formula and apply it to the candidate attribute classification.

VI. EXPERIMENTAL ANALYSIS

A. DATA SET AND EXPERIMENTAL ENVIRONMENT

In order to evaluate the proposed method, a data set [42] consisting of 1000 Chinese Mandarin texts was constructed based on the SOHU news corpus. It contains text data in the military, entertainment, sports, education and other categories. The attributes of the text resource in the dataset are pre-marked. After preprocessing and normalization of data such as stop words and invalid symbols, an experimental data set consisting of 19,600 pieces of attribute is finally obtained. The data set is randomly segmented to obtain a training set consisting of 15600 (80%) pieces of data, a verification set consisting of 2000 (10%) pieces of data, and a test set consisting of 2000 (10%) pieces of data. This paper uses the jieba software to implement text segmentation, and uses the Tensorflow-based neural network framework Keras to create and train the CNN network. The experimental software and hardware environment is as follows: the operating system is Win 10 64-bit, the CPU is Intel(R) Core(TM) i7-4710MQ@2.5GHz, the GPU is GeForce GTX 850M, the memory size is 16GB, and the Keras version is 2.1.3.

B. EVALUATION INDICATORS

The functional indicators of this method are mainly used to evaluate the mining effect of the candidate attributes. We define the confusion matrix of attribute classification results as follows:

TABLE 4. Confusion matrix of attribute classification results.

True category	Forecast category	
	Key attribute	Non-key attribute
Key attribute	N_{TP}	N_{FN}
Non-key attribute	N_{FP}	N_{TN}

where N_{TP} indicates the number of samples whose key attributes are correctly detected as key attributes; N_{FN} indicates the number of samples whose key attributes are incorrectly detected as non-key attributes; N_{FP} represents the number of samples whose non-key attributes are incorrectly detected as key attributes; and N_{TN} indicates the number of samples whose non-key attributes are correctly detected as non-key attributes. The corresponding evaluation indicators are as follows:

(1) Accuracy represents the proportion of the number of correct samples in the experimental results to the total number of samples. The formula is as follows:

$$acc = \frac{N_{TP} + N_{TN}}{N_{TP} + N_{TN} + N_{FP} + N_{FN}} \tag{14}$$

(2) Precision represents the proportion of the number of correct positive samples in the experimental results to the number of forecast positive sample. The formula is as follows:

$$pre = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{15}$$

(3) Recall represents the proportion of the number of correct positive samples in the experimental results to the number of actual positive samples. It is a measure of coverage, as follows:

$$re = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{16}$$

(4) F1-measure is the weighted harmonic average of the accuracy rate and recall rate. The formula is as follows:

$$F1 = \frac{2 * pre * re}{pre + re} \tag{17}$$

C. EXPERIMENTAL RESULTS

In order to evaluate the performance of this method, we designed the following four experiments: accuracy of attribute mining and evaluation of Loss value, comparison with other benchmark methods, evaluation of results of different training models under the same feature condition, and comparison of mining efficiency with different training models.

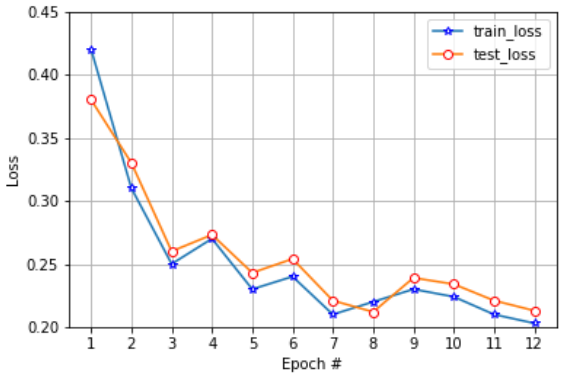


FIGURE 3. Accuracy of the training set and test set.

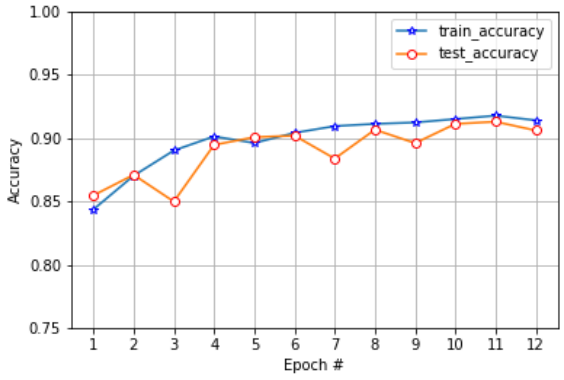


FIGURE 4. Loss values for training and test sets.

(1) Accuracy of attribute mining and evaluation of Loss value. As shown in fig.3 and fig.4, the trained attribute mining model can achieve an accuracy rate of 90.57% and a loss value of 0.213 with the test data set. This can basically meet the requirements of mining accuracy rate for accessing mining attributes of big data.

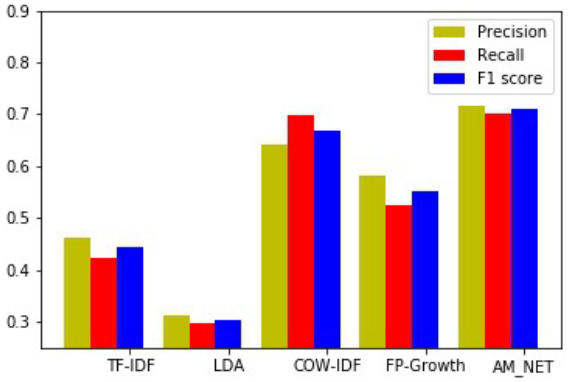


FIGURE 5. Comparison with other benchmark methods.

TABLE 5. Comparison with other benchmark methods.

	TF-IDF	LDA	FP-Growth	COW-IDF	AM_NET
Precision	0.461	0.312	0.582	0.643	**0.7171**
Recall	0.423	0.298	0.524	0.698	**0.7011**
F-measure	0.443	0.305	0.552	0.669	**0.7090**

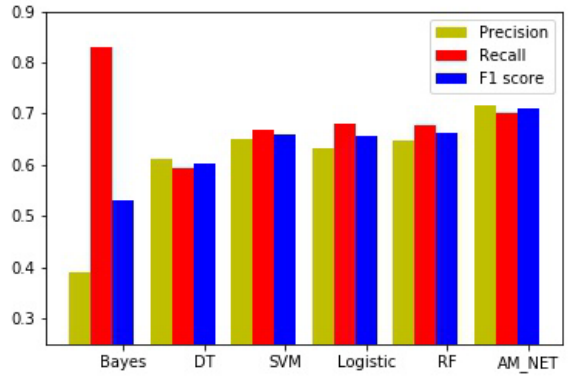


FIGURE 6. Comparison with different attribute mining classifiers.

(2) Comparison with other benchmark methods, namely TF-IDF, LDA, COW-IDF and FP-Growth. Fig. 5, it compares our method AM_NET with these other attribute mining methods for: the precision rate, the recall rate, and the F1-measure. It can be seen from the experimental results that our method AM-NET is superior in all three indicators.

(3) Evaluation of results of different training models under the same feature condition. Under the condition of selecting the same feature, the precision, recall rate and F1-measure of different attribute mining classifiers are compared. Fig. 6 shows that compared with naive Bayes, decision tree, support vector machine, logistic regression and random forest, the AM_NET method proposed in this paper has better overall performance. Whilst the precision rate and the F1-measure are best for AM_NET, the naive Bayesian model is best for the recall rate, but its other two indicators are poor.

(4) Comparison of mining efficiency with different training models. Table 7 compares AM_NET with the other

TABLE 6. Comparison with different attribute mining classifiers.

	Bayes	DT	SVM	Logistic	RF	AM_NET
Precision	0.3909	0.6129	0.6507	0.6340	0.6484	**0.7171**
Recall	**0.8310**	0.5949	0.6685	0.6797	0.6783	0.7011
F-measure	0.5317	0.6038	0.6595	0.6561	0.6630	**0.7090**

TABLE 7. Comparison of mining efficiency with different training models.

	Bayes	DT	SVM	Logistic	RF	AM_NET
Time (ms)	0.08	0.02	7.62	2.57	3.1	**24**

mining methods: Bayes, DT, SVM, Logistic and RF. AM_NET takes more time for single text detection than the other methods. This is mainly due to the CNN model having more parameters. Under our experimental conditions, the average time for the evaluation of a single text resource is about 24ms, which can meet the needs of real-time detection.

D. EXPERIMENTAL LIMITATIONS ANALYSIS

The experiments show that compared with existing classifiers, our AM_NET method shows a large (≈10%) improvement in precision, recall rate and F1-measure. Meanwhile, compared the existing benchmark methods (TF-IDF, LDA, COW-IDF and FP-Growth) the proposed scheme is shown to be the best. This is because we use more comprehensive multi-dimensional hybrid features to achieve accurate characterization of candidate attributes. And for this problem, convolutional neural network has a very strong learning ability. However, our experiments still have certain shortcomings. The word vector and text vector language models used in this paper are based on our experimental data sets. Therefore, its scalability is still to be verified with other data sets. Furthermore, in order to ensure the actual effect of attribute mining, it is necessary to construct a targeted language model for the processed text resources. This will adversely affect the performance improvement of attribute mining for different types of text resources. Since the experimental results are better for supervised algorithms, there is still a need to label the available data sets for training, which limits the rapid use and deployment of the method to some extent. Finally, the data in our test set was unbalanced data. This is because in most text resources, the number of general attributes is much larger than the number of key attributes. Therefore it still has to be tested if our method will perform better with more specialized texts.

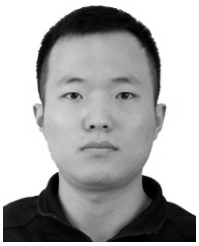
VII. CONCLUSION

In order to overcome the problems of traditional artificial attribute extraction management methods, such as high cost, time consuming, variable accuracy and poor scalability, this paper proposes a method based on a convolutional neural network for attribute mining from big data. It provides a new solution for the automation and intelligent mining and extraction of attributes. Such attributes might be used to support access control to unstructured big data text resources. First, we convert the candidate attributes into grayscale images. Afterwards, a CNN is used to identify and classify the image, thereby realizing the mining of text resource attributes. The experimental results show that our CNN can automatically analyze the features of the higher dimension of the extracted attributes and achieves better attribute mining results than existing methods, but at some cost to performance. In future work, we hope to achieve more accurate and efficient attribute mining based on text and attribute vectors. In addition, due to the limited amount of data in the data set of this paper, we will try to build a more complete data set to improve the next research results.

REFERENCES

[1] F. Liang, L. H. Yin, and Y. C. Guo, "A survey of key technologies in attribute-based access control scheme," *Chin. J. Comput.*, vol. 40, no. 7, pp. 1680–1698, 2017.
[2] X.-M. Wang, F. Hong, and L.-C. Zhang, "Research progress on attribute-based access control," *Acta Electron. Sinica*, vol. 38, no. 7, pp. 1660–1667, 2010.
[3] C. Yaokun, Y. Xianglan, and L. Wenli, "Access control model applicability for big data," *Inf. Secur. Technol.*, vol. 7, no. 7, pp. 3–5, 2016.
[4] J. Xin, R. Krishnan, and R. Sandhu, "A unified attribute-based access control model covering DAC, MAC and RBAC," in *Proc. 25th Data Appl. Secur. Privacy*, 2012, pp. 41–45.
[5] *Big Data*. Accessed: Jul. 2, 2019. [Online]. Available: https://baike.baidu.com/item/%E5%A4%A7%E6%95%B0%E6%8D%AE/1356941?fr=aladdin
[6] R. Sandhu, "The future of access control: Attributes, automation, and adaptation," in *Proc. IEEE Int. Conf. Inf. Reuse Integr.*, 2014, p. 45.
[7] H. Li, M. Zhang, D. G. Feng, and Z. Hui, "Research on access control of big data," *Chin. J. Comput.*, vol. 40, no. 1, pp. 72–91, 2017.
[8] W. Zeng, Y. Yang, and L. Bo, "Access control for big data using data content," in *Proc. IEEE Int. Conf. Big Data*, Oct. 2013, pp. 45–47.
[9] D. G. Feng, M. Zhang, and L. I. Hao, "Big data security and privacy protection," *Chin. J. Comput.*, vol. 37, no. 1, pp. 246–258, 2014.
[10] Y. Jung and J. B. D. Joshi, "CPBAC: Property-based access control model for secure cooperation in online social networks," *Comput. Secur.*, vol. 41, no. 3, pp. 19–39, 2014.
[11] J. Zhang, A. Castiglione, L. T. Yang, and Y. Zhang, "Recent advances in security and privacy in social big data," *Future Gener. Comput. Syst.*, vol. 87, pp. 686–687, Oct. 2018.
[12] H. A. Al Hamid, S. M. M. Rahman, M. S. Hossain, A. Almogren, and A. Alamri, "A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography," *IEEE Access*, vol. 5, pp. 22313–22328, 2017.
[13] Z. Hui, H. Li, M. Zhang, and D.-G. Feng, "Risk-adaptive access control model for big data in healthcare," *J. Commun.*, vol. 36, no. 12, pp. 190–199, 2015.
[14] I. Molloy and S. Chari, "Generative models for access control policies: Applications to role mining over logs with attribution," in *Proc. ACM Symp. Access Control Models Technol.*, 2012, pp. 45–46.
[15] L. Yin, F. Liang, B. Niu, B. Fang, and F. Li, "Hunting abnormal configurations for permission-sensitive role mining," in *Proc. Military Commun. Conf.*, Nov. 2016, pp. 1004–1009.
[16] T. Lee, Z. Wang, H. Wang, and S. W. Hwang, "Attribute extraction and scoring: A probabilistic approach," in *Proc. IEEE Int. Conf. Data Eng.*, Aug. 2013, pp. 194–205.
[17] Y. Li, B. J. P. Hsu, C. X. Zhai, and K. Wang, "Mining entity attribute synonyms via compact clustering," in *Proc. ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2013, pp. 867–872.
[18] H. J. Ming, "Research and implementation on public opinion analysisand attribute discovery orientied Internet text mining," Ph.D. dissertation, Dept. Elect. Eng., Nat. Univ. Defense Technol., Changsha, China, 2011.

[19] G.-Q. Hu and J.-H. Li, "A Credibility Analysis-Based Method to Discover New Attributes Web Pages," *Comput. Technol. Develop.*, vol. 19, no. 1, pp. 56–59, 2009.
[20] J. S. Zhao, Q. M. Zhu, G. D. Zhou, and L. Zhang, "Review of research in automatic keyword extraction," *J. Softw.*, vol. 28, no. 9, pp. 2431–2449, 2017.
[21] Y. C. Chang, Y. X. Zhang, H. Wang, H. Y. Wan, and C. J. Xiao, "Features oriented survey of state-of-the-art keyphrase extraction algorithms," *J. Softw.*, vol. 29, no. 7, pp. 1–25, 2018.
[22] K. Hofmann, M. Tsagkias, E. Meij, and M. D. Rijke, "The impact of document structure on keyphrase extraction," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 1725–1728.
[23] R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2004, pp. 404–411.
[24] Y. Ma, K. Yue, Z. C. Zhang, X. Y. Wang, and J. B. Guo, "Extraction of social media databased on the knowledge graph and LDA model," *J. East China Normal Univ.(Natural Sci.)*, vol. 201, no. 5, pp. 183–194, 2018.
[25] L. Huang, Y. P. Wu, and Q. F. Zhu, "Research and improvement of TFIDF text feature weighting method," *Comput. Sci.*, vol. 41, no. 6, pp. 204–207, 2014.
[26] L. Liu and T. Peng, "Clustering-based method for positive and unlabeled text categorization enhanced by improved TFIDF," *J. Inf. Sci. Eng.*, vol. 30, no. 5, pp. 1463–1481, 2014.
[27] H. F. Ma, F. Liu, Q. Xia, and Z. J. Hao, "Keywords extraction algorithm based on weighted hypergraph random walk," *Acta Electron. Sinica*, vol. 46, no. 6, pp. 1410–1414, 2018.
[28] C. Florescu and C. Caragea, "A position-biased pagerank algorithm for keyphrase extraction," in *Proc. Nat. Conf. Artif. Intell.*, 2017, pp. 4923–4924.
[29] F. Boudin, "Reducing over-generation errors for automatic keyphrase extraction using integer linear programming," in *Proc. Meeting Assoc. Comput. Linguistics*, 2015, pp. 19–25.
[30] F. Boudin, "A comparison of centrality measures for graph-based keyphrase extraction," in *Proc. Int. Joint Conf. Natural Language Process.*, 2013, pp. 834–838.
[31] R. Saeidi, R. F. Astudillo, and D. Kolossa, "Uncertain LDA: Including observation uncertainties in discriminative transforms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1479–1488, Jul. 2016.
[32] M. Haddoud and A. Said, "Accurate keyphrase extraction by discriminating overlapping phrases," *J. Inf. Sci.*, vol. 40, no. 4, pp. 488–500, 2014.
[33] C. Caragea, F. Bulgarov, A. Godea, and S. D. Gollapalli, " Citation-enhanced keyphrase extraction from research papers: A supervised approach," in *Proc. Empirical Methods Natural Language Process.*, 2014, pp. 1435–1446.
[34] S. D. Gollapalli, X. Li, and P. Yang, "Incorporating expert knowledge into keyphrase extraction," in *Proc. Nat. Conf. Artif. Intell.*, 2017, pp. 3180–3187.
[35] R. Meng, S. Zhao, S. Han, D. He, P. Brusilovsky, and Y. Chi, "Deep keyphrase generation," in *Proc. Meeting Assoc. Comput. Linguistics*, 2017, pp. 582–592.
[36] L. Sterckx, C. Caragea, and T. Demeester, "Supervised keyphrase extraction as positive unlabeled learning," in *Proc. Empirical Methods Natural Language Process.*, 2016, pp. 1924–1929.
[37] A. K. John, L. D. Caro, and G. Boella, "A supervised key phrase extraction system," in *Proc. Int. Conf. Semantic Syst.*, 2016, pp. 57–62.
[38] Y. Chen, R. Zhou, W. Zhu, M. T. Li, and J. Yin, "Mining patent knowledge for automatic keyword extraction," *J. Comput. Res. Develop.*, vol. 53, no. 8, pp. 1740–1752, 2016.
[39] *Keyword Extraction*. Accessed: Oct. 24, 2018. [Online]. Available: https://github.com/bigzhao/Keyword_Extraction
[40] Z. Cui, X. Fei, X. Cai, C. Yang, G. G. Wang, and J. Chen, "Detection of malicious code variants based on deep learning," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3187–3196, Jul. 2018.
[41] A. G. Howard *et al.*. (2017). "MobileNets: Efficient convolutional neural networks for mobile vision applications." [Online]. Available: https://arxiv.org/pdf/1704.04861.pdf
[42] *Data Set*. Accessed: Feb. 24, 2019. [Online]. Available: https://pan.baidu.com/s/19hdsSCbq3Xu51iScjxn-jg



AODI LIU received the B.S. and M.S. degrees from the Zhengzhou Science and Technology Institute, Zhengzhou, China, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree. His research interests include big data security and network security.

XUEHUI DU received the Ph.D. degree from the Zhengzhou Science and Technology Institute, Zhengzhou, China, in 2012, where she is currently a Professor. Her research interests include cloud computing and big data security.

NA WANG received the B.S., M.S., and Ph.D. degrees from the Zhengzhou Science and Technology Institute, Zhengzhou, China, in 2001, 2004, and 2008, respectively, where she is currently an Associate Professor. Her research interests include cloud computing and trust management.