Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.059784

ARTICLE

Institution Attribute Mining Technology for Access Control Based on Hybrid Capsule Network

Aodi Liu, Xuehui Du*, Na Wang and Xiangyu Wu

Henan Province Key Laboratory of Information Security, Information Engineering University, Zhengzhou, 450000, China

*Corresponding Author: Xuehui Du. Email: dxh37139@163.com

Received: 16 October 2024; Accepted: 14 January 2025; Published: 26 March 2025

ABSTRACT: Security attributes are the premise and foundation for implementing Attribute-Based Access Control (ABAC) mechanisms. However, when dealing with massive volumes of unstructured text big data resources, the current attribute management methods based on manual extraction face several issues, such as high costs for attribute extraction, long processing times, unstable accuracy, and poor scalability. To address these problems, this paper proposes an attribute mining technology for access control institutions based on hybrid capsule networks. This technology leverages transfer learning ideas, utilizing Bidirectional Encoder Representations from Transformers (BERT) pre-trained language models to achieve vectorization of unstructured text data resources. Furthermore, we have designed a novel end-to-end parallel hybrid network structure, where the parallel networks handle global and local information features of the text that they excel at, respectively. By employing techniques such as attention mechanisms, capsule networks, and dynamic routing, effective mining of security attributes for access control resources has been achieved. Finally, we evaluated the performance level of the proposed attribute mining method for access control institutions through experiments on the medical referral text resource dataset. The experimental results show that, compared with baseline algorithms, our method adopts a parallel network structure that can better balance global and local feature information, resulting in improved overall performance. Specifically, it achieves a comprehensive performance enhancement of 2.06% to 8.18% in the F1 score metric. Therefore, this technology can effectively provide attribute support for access control of unstructured text big data resources.

KEYWORDS: Access control; ABAC model; attribute mining; capsule network; deep learning

1 Introduction

In recent years, the rapidly evolving big data technology has increasingly attracted widespread attention from all sectors of society. While big data provides significant convenience for production and daily life, it also brings substantial data security risks. Effectively ensuring the security and controllability of big data resources has become a prerequisite for big data sharing and its applications [1]. As one of the core technologies for data security management, access control technology [2,3] can prevent data resources from being accessed and used without authorization by managing user permissions, thereby achieving secure protection of big data resources. Among these, the Attribute-Based Access Control (ABAC) mechanism [4], which uses attributes as fundamental elements of access control, can flexibly determine whether to grant users related access permissions based on the set of attributes possessed by entities. Compared with traditional access control mechanisms such as Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Role-Based Access Control (RBAC) [5], ABAC offers advantages such as strong semantic expression capability, high policy flexibility, and efficient permission decision-making, making it suitable for addressing fine-grained access control and large-scale dynamic authorization issues in big data environments.

Access control attributes are the foundation of the ABAC [6] (Attribute-Based Access Control) mechanism, including subject attributes, resource attributes, action attributes, and environmental attributes. Existing attribute management methods based on expert knowledge can effectively manage subject attributes, action attributes, and environmental attributes, which have relatively limited scales. However, for unstructured big data resources [7,8] characterized by large data volumes, rapid dynamic generation, and diverse data types, it is challenging for security experts to achieve the labeling and management of massive resource attributes. There is an urgent need to realize automated mining of access control attributes to provide effective attribute support for implementing attribute-based access control over vast resources [9].

Institutional attributes are a type of resource attribute information used to describe the management and application institutions to which unstructured text resources belong. During the implementation phase of access control, institutional attributes can impose spatial constraints on the access control behavior of data resources, making them an important category of access control resource attributes. In access control policies, institutional attributes can be utilized to authorize and control user access behaviors. For example, entities that initiate access requests can access data resources belonging to corresponding institutions. Therefore, tagging object resources with institutional attributes holds significant value for access control. Given that text data represents a typical form of unstructured data resources, this paper focuses on the problem of mining institutional attributes from unstructured text data resources. It transforms the issue of mining institutional attributes into a multi-institutional attribute tagging problem, aiming to accurately describe the institutions to which unstructured text resources belong and incorporate the institutional attribute information of resources into the process of implementing access control. This approach aims to provide institutional attribute support for efficient and fine-grained access control.

The main contributions of this paper include: transforming the problem of mining institutional attributes from unstructured text data resources into a natural language processing problem; proposing an institutional attribute mining method for access control based on hybrid capsule networks, named ATT_BiLSTM_Capsule_Net; designing a novel end-to-end parallel neural network structure capable of integrating both global and local feature information of text resources; leveraging transfer learning concepts to utilize BERT pre-trained language models for the vectorization of unstructured text data resources; and employing techniques such as attention mechanisms, capsule networks, and dynamic routing to achieve effective mining of institutional attributes for access control. Experimental results show that, in terms of institutional attribute mining, compared with existing benchmark methods, our approach achieves a comprehensive performance improvement ranging from 2.06% to 8.18% in the F1 score metric. This indicates that the proposed method can provide effective auxiliary intelligent decision support for attribute management in big data access control.

2 Related Work

The quality of security attribute settings directly impacts the effectiveness of access control. Currently, in the field of access control, attribute management primarily relies on expert knowledge, and direct research on security attribute mining specifically for access control is still relatively scarce [10,11]. This study focuses on the issue of attribute mining for unstructured text data resources, where text keywords, topics, and categories are all crucial information for describing and characterizing these resources. Techniques from the field of natural language processing, such as keyword extraction and multi-label tagging, offer valuable insights and are relevant to this research. The comparison of related methods is shown in Table 1.

Table 1: Comparison of different methods

Methods	Advantage	Disadvantage
Statistical methods	Simple and easy to implement, high computational efficiency.	The relationship between words and context information are ignored, and the semantic understanding ability is limited.
Graph-based methods	The correlation between words can be captured, which can effectively simulate human reading habits.	The calculation complexity is high, and the construction and weight setting of the graph will affect the quality of the result.
Topic modeling methods	The potential distribution of topics can be found in the document, providing a deeper understanding.	It takes a lot of time and computing resources. The results of the model are poorly interpreted and difficult for users to understand.
Classifier-based methods	Use feature engineering to incorporate more contextual information and have greater accuracy.	It is difficult to focus on global and local context information at the same time, and the cost of migrating to new domains is high.
Sequence labeling-based methods	For short texts and text in specific formats, it can handle the sequential nature of natural language well.	For longer text, performance may be affected due to context window restrictions.

(1) Keyword Extraction Technology Based on Natural Language Processing

Current research can be categorized into unsupervised and supervised keyword extraction techniques based on the different methods employed. The mainstream unsupervised keyword extraction methods can be summarized into three categories along with their respective improvements: statistical methods, graph-based methods, and topic modeling methods. Statistical Methods [12] assess the importance of candidate words by utilizing statistical indicators derived from text resources to extract keywords from the text. Graph-Based Methods [13,14] transform text resources into directed or undirected language structure graphs. By analyzing the structure of the graph, key nodes that play a crucial role are identified and treated as keywords of the text. Topic Modeling Methods [15,16] construct probabilistic language models representing the relationship between texts and topics, as well as between topics and keywords, using probabilistic methods to infer keywords. Unsupervised methods do not require the construction of annotated text training datasets or the training of extraction models. They can obtain keywords within texts by ranking candidate words, featuring strong generalization capabilities and versatility. However, these methods suffer from issues such as poor extraction performance and instability. On the other hand, supervised methods primarily transform the keyword extraction problem into binary classification problems or text annotation problems in machine learning. These methods require model training based on corpora annotated by humans, utilizing trained machine learning models to extract keywords from new text resources. Commonly used supervised extraction techniques leverage Support Vector Machine (SVM) algorithms [17], Neural Network algorithms [18], and others to achieve keyword extraction. Supervised methods mainly follow two development directions: one direction is to convert keyword extraction into a binary classification problem determining whether candidate words are keywords; the other direction involves constructing text annotation models, using natural language processing techniques such as sequence labeling [19,20] to achieve keyword extraction.

(2) Multi-Label Tagging Technology Based on Natural Language Processing

With the development and popularization of deep learning, deep learning technologies have been widely applied to research in the field of multi-label tagging. Kim et al. [21] initially applied Convolutional Neural Networks (CNN) models to text classification, directly using convolutional networks for sentence-level text, but using convolutional networks alone for text classification lacks analysis of the temporal characteristics of text. Peng et al. [22] introduced hierarchical classification awareness mechanisms into the network to explore the cross-layer semantic associations in text. Due to the temporal nature of text resources, using Recurrent Neural Network (RNN) networks for text classification is more straightforward and simpler; however, in practical research, the performance of single RNN networks was found not to be as successful as expected [23]. With the development of technologies such as capsule networks, sequence autoencoder initialization, and embedding perturbation information in neural networks [24,25], network models have been able to achieve better classification and tagging performance. Additionally, self-attention networks [26] without any convolutional or recurrent models have also been successfully applied to multi-label tagging. Since the convolution operators of CNNs are represented by weighted sums at the lower level, it becomes difficult to express the features of complex objects as they move to higher layers. The drawback of this approach is that it does not consider the hierarchical relationships between local features. CNN networks can overcome these drawbacks by utilizing pooling, which reduces the computational complexity of convolution operations and captures the invariance of local features. However, pooling operations can also lose feature information about spatial relationships and may misclassify objects based on their orientation or scale. Subsequently, capsule networks [27,28] were proposed to address the inherent issues of CNN networks by learning to recognize the presence of visual entities and encoding their attributes as vectors of local invariants. Using capsule networks can effectively extract local features rich in linguistic information. However, due to the strong correlation of contextual information in text resources, using a single capsule network lacks observation and extraction of global features, limiting further performance improvements.

From the above analysis, it can be seen that the research on mining security attributes of unstructured text data is still in the exploratory stage, but advancements can be promoted by drawing on natural language processing technologies. In existing similar studies, unsupervised methods only consider structural features, leading to poor overall extraction performance. Supervised methods, on the other hand, require large amounts of manually annotated corpora for training, which presents shortcomings such as poor robustness and weak scalability. Moreover, the current techniques extract semantic information that is not directly related to access control. Further research is needed to explore how to combine the specific needs of access control for unstructured text data, targeting the mining and optimization of security attributes.

3 Institution Attribute Mining Model

3.1 Task Description

The Attribute-Based Access Control (ABAC) model is an access control method based on attributes, controlling users' access to resources through the management of these attributes. As shown in Fig. 1, the entire lifecycle of attribute management in the ABAC model consists of six stages: attribute definition, attribute assignment, attribute evaluation, attribute authorization, attribute revocation, and attribute auditing. Together, these stages ensure that users' access to resources is effectively controlled and managed.

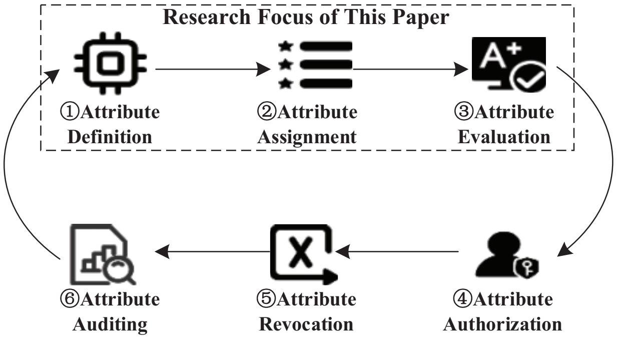


Figure 1: Full lifecycle management of access control attributes

Attribute Definition: This stage is primarily responsible for defining and describing access control attributes. Attributes are used to represent the characteristics of entities such as users, resources, and environments, including user identity, resource type, affiliated institution, and access time, among others. Attribute definition includes specifying the name, type, and value range of the attribute.

Attribute Assignment: This stage is responsible for assigning attribute values to entities such as users, resources, and environments. Attribute assignment can be performed automatically based on predefined rules, user requests, or external data sources.

Attribute Evaluation: This stage evaluates whether the assigned attribute values for users, resources, etc., are appropriate according to access control security requirements.

Attribute Authorization: Based on the results of attribute evaluation, this stage assigns the corresponding attributes to users and grants access permissions. Attribute authorization can be carried out automatically based on predefined policies, user requests, or external data sources.

Attribute Revocation: This stage is responsible for revoking access permissions that users have already obtained. Attribute revocation can be performed automatically based on user requests, administrator actions, or external data sources. For example, when a user leaves the organization or changes positions, their original attributes can be revoked.

Attribute Auditing: This stage records and tracks users' access activities to resources. Attribute auditing helps organizations understand the actual usage of resources by users, facilitating security analysis and incident investigations.

The focus of this paper on the mining of institutional attributes primarily centers on stages ①, ②, and ③ depicted in Fig. 1: defining institutional attributes, assigning institutional attributes to resources, and evaluating the effectiveness of attribute assignments. The problem of mining institutional attributes is defined as follows: Let $R = \{r_1, r_2, \ldots, r_n\}$ be a set of n unstructured text resources that require access control protection. Each resource $r_i \in R$ should have a unique corresponding institutional attribute A_I. The goal of institutional attribute mining is:

(1) To define a set of candidate institutional attributes $IC_i = \{ic_{i,1}, ic_{i,2}, \ldots, ic_{i,k}\}$ for all resources.

(2) To find a transformation function T, which maps the unstructured text resource R to the corresponding institutional attribute A_I in the candidate institutional attribute set IC_i.

3.2 Network Model Architecture

This paper adopts the ATT_BiLSTM_Capsule_Net combined network model to address the issue of mining institutional attributes of unstructured text resources. As shown in Fig. 2, the model comprises three layers: the input layer, the joint processing layer, and the fusion output layer. The input layer leverages the concept of transfer learning, converting input text resources into word vectors as feature outputs through the Bidirectional Encoder Representations from Transformers (BERT) pre-trained language model. The output features are then fed into the ATT_BiLSTM layer and the Capsule_Net layer for parallel computation and analysis. In the ATT_BiLSTM layer, features first pass through a SpatialDropout1D layer to prevent overfitting of the network model. Subsequently, an Attention mechanism layer is established, followed by a Bidirectional Long Short-Term Memory (BiLSTM) layer to acquire and mine the global semantic relationships and feature information within the text resources, producing the corresponding feature processing result *FPR1*. Meanwhile, in the Capsule_Net layer, parallel feature calculation is performed, starting with a Conv1D layer to obtain the contextual relationships of text resources within the receptive field, followed by a GatedConv unit to activate local effective features. The Capsule layer then computes the local feature information of text resources using dynamic routing mechanisms, generating the feature processing result *FPR2*. Finally, the fusion output layer concatenates the output results *FPR1* and *FPR2* from the ATT_BiLSTM layer and the Capsule_Net layer, achieving a mixed feature fusion of global and local information of the text resources to be mined. The fused features are input into a fully connected network (Dense), and the Softmax function is used to obtain the institutional attribute information corresponding to the resource, thus realizing the mining of institutional attributes of unstructured text resources in the context of access control applications. Below, we will separately introduce each sub-network and the overall process of combining sub-networks for parallel analysis and learning.

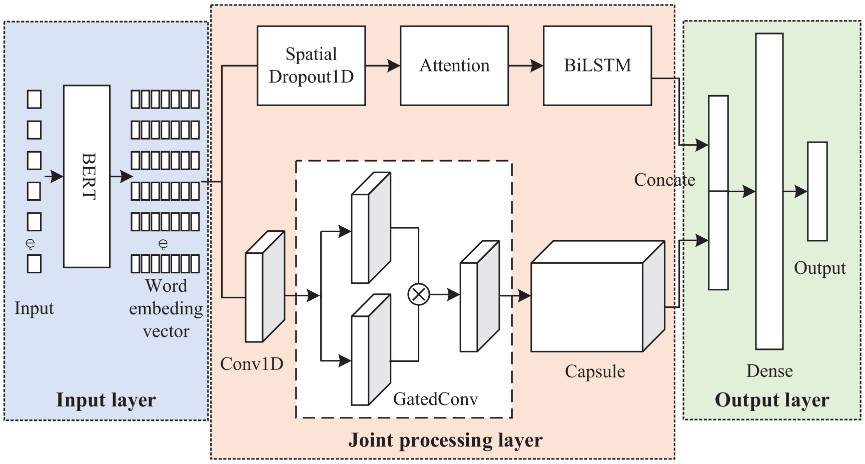


Figure 2: ATT_BiLSTM_Capsule_Net model

3.3 Input Layer

In the input layer, we use the BERT (Bidirectional Encoder Representations from Transformers) pre-trained language model [29], developed by Google, to perform vectorization of the input unstructured text resources, serving as the input for subsequent models. The BERT model shifts the extensive word vectorization operations traditionally performed in downstream specific Natural Language Processing (NLP) tasks to the pre-trained language model. Based on the concept of transfer learning, the pre-trained language model, trained on massive text resources across various domains, is applied to solve the problem of mining institutional attributes of unstructured text resources for access control. This approach effectively avoids the issue of insufficient specific training data in particular NLP tasks, enhancing the generalization capability of word vector models. It adequately describes the representation features at the character, word, and sentence levels.

As shown in Fig. 3, the structural characteristics of different language models highlight that the BERT model trains word vectors using bidirectional transformer technology. Compared with other language model structures, the BERT model integrates the advantages of models such as OpenAI GPT and ELMo [30], capable of establishing contextual dependencies from both sides simultaneously. It also features deeper training layers and better training parallelism, demonstrating excellent performance in various NLP tasks.

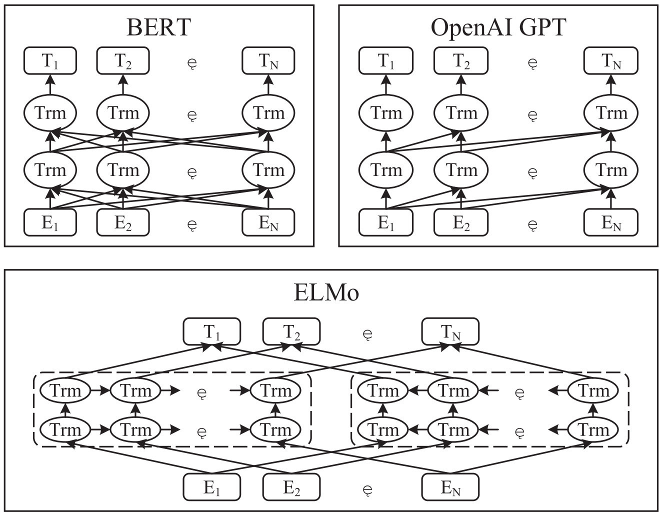


Figure 3: Structure of different language models

The process of vectorizing unstructured text resources using the BERT model is illustrated in Fig. 4. Token Embeddings are used to represent word vectors, which can be either word vectors or character vectors when processing Chinese text resources. To achieve fine-grained text representation, this paper uses character vectors to describe the resources. The classification (CLS) marker is the first character of the text resource, used to denote the beginning of the text resource. The separator (SEP) marker is used to indicate the end and segmentation of sentences within the text resource. Segment Embeddings are used to distinguish information from different sentences. Position Embeddings are used to learn and obtain positional information of characters and words within the text. For a given text resource, after summing these three parts of embeddings, the final text-level vector representation of the input text resource is obtained.

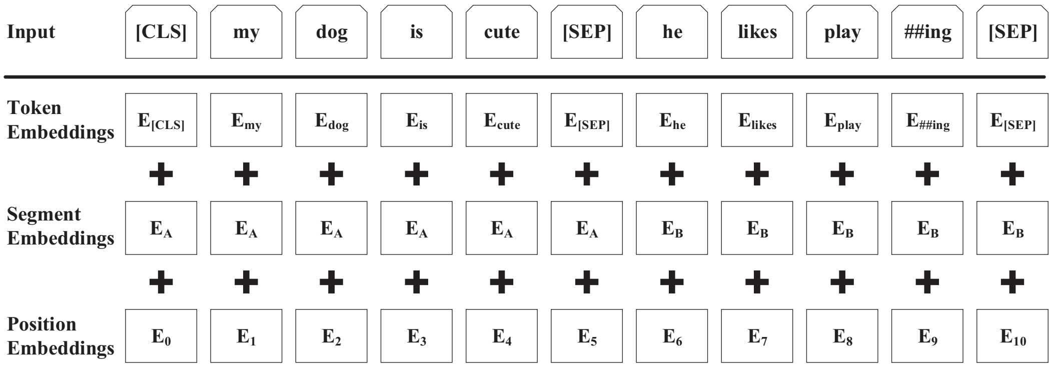


Figure 4: Input and output of BERT model

3.4 Joint Processing Layer

(1) ATT_BiLSTM Layer

Receiving the word vector features of text resources output from the input layer, this layer outputs the feature processing result *FPR1*. First, the attention weights between different word vector features are calculated, using the following method:

$$e_i = \tanh\left(w_i \cdot h_i + b_i\right) \tag{1}$$

$$\alpha_i = \frac{\exp\left(e_i\right)}{\sum_{i=1}^{n} \exp\left(e_i\right)} \tag{2}$$

here, h_i is the i-th hidden state input to the attention mechanism, w_i is the weight coefficient, and b_i is the bias. a_i is the attention weight corresponding to h_i. According to the attention weight vector $\alpha \in R^L$, a weighted calculation of h is performed to obtain the output s after the attention mechanism, calculated as follows:

$$s = \sum_{i=1}^{t} \alpha_i h_i \tag{3}$$

Considering the contextual relevance of words in text resources, a word may be associated with both its preceding and succeeding words. Therefore, the BiLSTM network is introduced, linking two Long Short-Term Memory (LSTM) with opposite temporal directions into the same network output, allowing the model to capture both historical and future information. Each LSTM structural unit in a BiLSTM includes four components: the input gate i, forget gate f, output gate o, and cell state c, with the calculation methods as follows:

$$\begin{aligned}
i_{(t)} &= \sigma\left(W_i x_{(t)} + u_i h_{(t-1)} + b_i\right) \\
f_{(t)} &= \sigma\left(W_f x_{(t)} + u_f h_{(t-1)} + b_f\right) \\
o_{(t)} &= \sigma\left(W_o x_{(t)} + u_o h_{(t-1)} + b_o\right) \\
g_{(t)} &= \tanh\left(W_g x_{(t)} + u_g h_{(t-1)} + b_g\right) \\
c_{(t)} &= f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)} \\
h_{(t)} &= o_t \otimes \tanh\left(c_t\right)
\end{aligned} \tag{4}$$

here, $i_{(t)}, f_{(t)}$, $o_{(t)}$ and $c_{(t)}$ represent the values of the input gate, forget gate, output gate, and cell state at time t, respectively. $x_{(t)}$ represents the input word vector at time t, and $h_{(t)}$ represents the hidden layer vector at time t. σ denotes the sigmoid activation function, while W and b are the weight matrix and bias vector, respectively. $h_{forward}$ and $h_{backward}$ are the hidden layer vectors output by the structural units of the forward LSTM and backward LSTM in the BiLSTM, respectively.

Concatenating $h_{forward}$ and $h_{backward}$ yields the output of the BiLSTM at time t, calculated as follows:

$$C_t = concat\left(h_{forward}, h_{backward}\right) \tag{5}$$

here, $h_{forward}$ and $h_{backward}$ correspond to the contextual information in two directions of the unstructured text resource, respectively.

(2) Capsule_Net Layer

Receiving the word vector features of text resources output from the input layer, this layer outputs the feature processing result *FPR2*. First, one-dimensional standard convolution calculations are performed on the word vector features of text resources. This layer extracts N-gram features at different positions in the sentence through various convolutional filters. The calculation method is as follows:

$$a_j^{(l)} = f\left(u_j^{(l)}\right) = f\left(\sum_{i \in N_j} a_j^{(l-1)} * K_{i,j}^{(l)} + b_j^{(l)}\right) \tag{6}$$

here, N_j is the set of input mappings, $K_{i,j}^{(l)}$ is the convolution kernel used to connect the i-th input feature map and the j-th output feature map, $b_j^{(l)}$ is the bias term of the j-th feature map, and f is the activation function.

Gated Linear Units (GLUs) can effectively reduce the occurrence of gradient vanishing phenomena while preserving the non-linear computational capabilities of the computational units. GLUs introduce a gating mechanism in convolutional calculations to control which features can be activated. Compared to pooling operations, GLUs do not lose spatial information. The calculation method is as follows:

$$g\left(X\right) = \left(X * W + b\right) \otimes \sigma\left(X * V + c\right) \tag{7}$$

The Capsule Network differs from Convolutional Neural Networks (CNNs) in its output form. While the output of a CNN is a scalar, the output of a Capsule Network is a vector. Capsule Networks can better handle the ambiguity of different types of data, enabling simultaneous recognition of various text structural patterns, such as the positional information of words and their syntactic structures. Therefore, we introduce Capsule Networks to handle the task of mining institutional attributes, extracting local features rich in contextual linguistic information.

As shown in Fig. 5, in Capsule Networks, the squashing compression function is used as the network's activation function to compress vectors. This maintains the direction of the vector input while scaling the length of the input vector to between (0, 1). The non-linear squashing function can be viewed as a method for compressing and redistributing the input vector, with the calculation method as follows:

$$v_j = squash\left(s_j\right) = \frac{s_j}{\left\|s_j\right\|} \cdot \frac{\left\|s_j\right\|^2}{1 + \left\|s_j\right\|^2} \tag{8}$$

here, v_j is the output vector of the j-th capsule, and s_j is the total input vector.

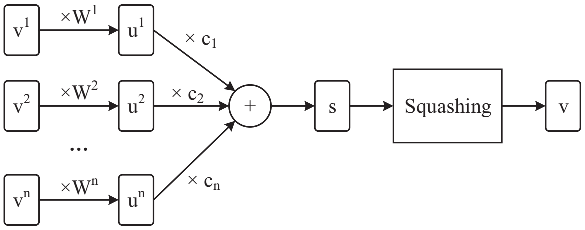


Figure 5: Dynamic routing mechanism

The Capsule Network calculates the coupling coefficients between capsules in different layers through the Dynamic Routing mechanism, which is akin to performing clustering calculations on input feature vectors, thereby undergoing a process of feature selection. According to the dynamic routing mechanism, when clustering features, the more similar features there are for a certain feature, the stronger that feature category is considered to be. This effectively diminishes outlier features and selects features with strong expression capabilities. The detailed calculation method is as follows:

$$c_{ij} = \frac{\exp\left(b_{ij}\right)}{\sum_k \exp\left(b_{ik}\right)} \tag{9}$$

here, c_{ij} is the coupling coefficient between capsules in different layers, representing the corresponding weight between lower-level capsules and higher-level capsules. b_{ij} is the connection weight between capsule i and capsule j, determined by the similarity between the capsules, used to predict the output vector of the upper capsule.

The total input vector s_j for all capsules is the weighted sum of all prediction vectors from the lower capsule layer, calculated as follows:

$$\begin{aligned} \hat{u}_{j|i} &= W_{ij}u_i \\ s_j &= \sum_i c_{ij}\hat{u}_{j|i} \end{aligned} \tag{10}$$

here, $\hat{u}_{j|i}$ is the prediction vector from the lower capsule, W_{ij} is the weight matrix, and u_i is the output vector of the upper capsule.

The b_{ij}, as the connection weight between capsule i and capsule j, is used to update the coupling coefficient c_{ij}. During the dynamic routing iterative calculation process, b_{ij} will be continuously updated. The calculation method is as follows:

$$\begin{aligned} v_j &= squash\left(s_j\right) \\ b_{ij} &= b_{ij} + \hat{u}_{j|i} \cdot v_j \end{aligned} \tag{11}$$

The number of capsules and the number of dynamic routing iterations are two crucial parameters in capsule networks, and their selection has a critical impact on model performance. The number of capsules directly determines the number of feature dimensions or abstract concepts that the model can learn. A greater number of capsules means the model can capture more complex patterns and structures, which is highly beneficial for handling text classification tasks with rich semantic information. However, as the number of capsules increases, so does the computational complexity of the model. Each capsule contains a set of neurons used to represent different aspects of specific attributes, meaning that more capsules lead to higher computational costs. If there are too many capsules without sufficient training data to support such extensive feature learning, it may result in overfitting. In this case, the model might perform well on the training set but exhibit poor generalization on unseen data. Typically, cross-validation and other experimental methods are needed to find an optimal number of capsules that ensures good model performance without excessively increasing computational burden. It is generally recommended to start with 5 to 25 capsules and experiment to find the best configuration.

Dynamic routing is a key mechanism in capsule networks that determines the connection weights between lower-level and higher-level capsules. More iterations of dynamic routing can make information transmission more accurate because each iteration updates these weights, allowing the model to better match the relationship between input data and target outputs. However, too many iterations can prolong the training time and may not necessarily lead to significant performance improvements. In some cases, excessive iterations can cause the model to get stuck in local optima, negatively impacting the final classification results. An appropriate number of iterations helps enhance the stability of the model, avoiding misclassification due to insufficient information transmission. Too few iterations may prevent the model from fully learning the complex structure of the input data. It is usually advised to start with 4 to 8 iterations, which is often a reasonable range. Based on this, the number of iterations can be adjusted according to the requirements of the actual task and available computational resources. During the training process, it is important to closely monitor the model's performance, including metrics such as training error, validation error, and convergence speed. If the model converges well and performs satisfactorily with fewer iterations, there is no need to increase the number of iterations; conversely, if the model converges slowly or shows signs of underfitting, it may be worth considering an increase in the number of iterations.

3.5 Output Layer

This layer fuses the feature processing results *FPR1* and *FPR2* obtained from the joint processing layer. Specifically, the operation involves concatenating *FPR2* to *FPR1* to obtain the fused feature *FF*. The calculation method is as follows:

$$FF = FPR1 \oplus FPR2 \tag{12}$$

here, $\oplus$ denotes the concatenation operation.

Subsequently, the fused feature *FF* is input into a fully connected layer (Dense) to obtain the output *f*, and then the Softmax function is applied to get the result *S*(*f*), which calculates the probability that the institutional attribute of the input text data resource is *k*. The Softmax calculation method is as follows:

$$S_k\left(f\right) = \frac{\exp\left(f\right)}{\sum_K \exp\left(f\right)} \tag{13}$$

here, K represents the total number of categories, and $S_k\left(f\right)$ is the probability that the institutional attribute is k.

4 Experiment and Analysis

4.1 Data Sets and Experimental Environments

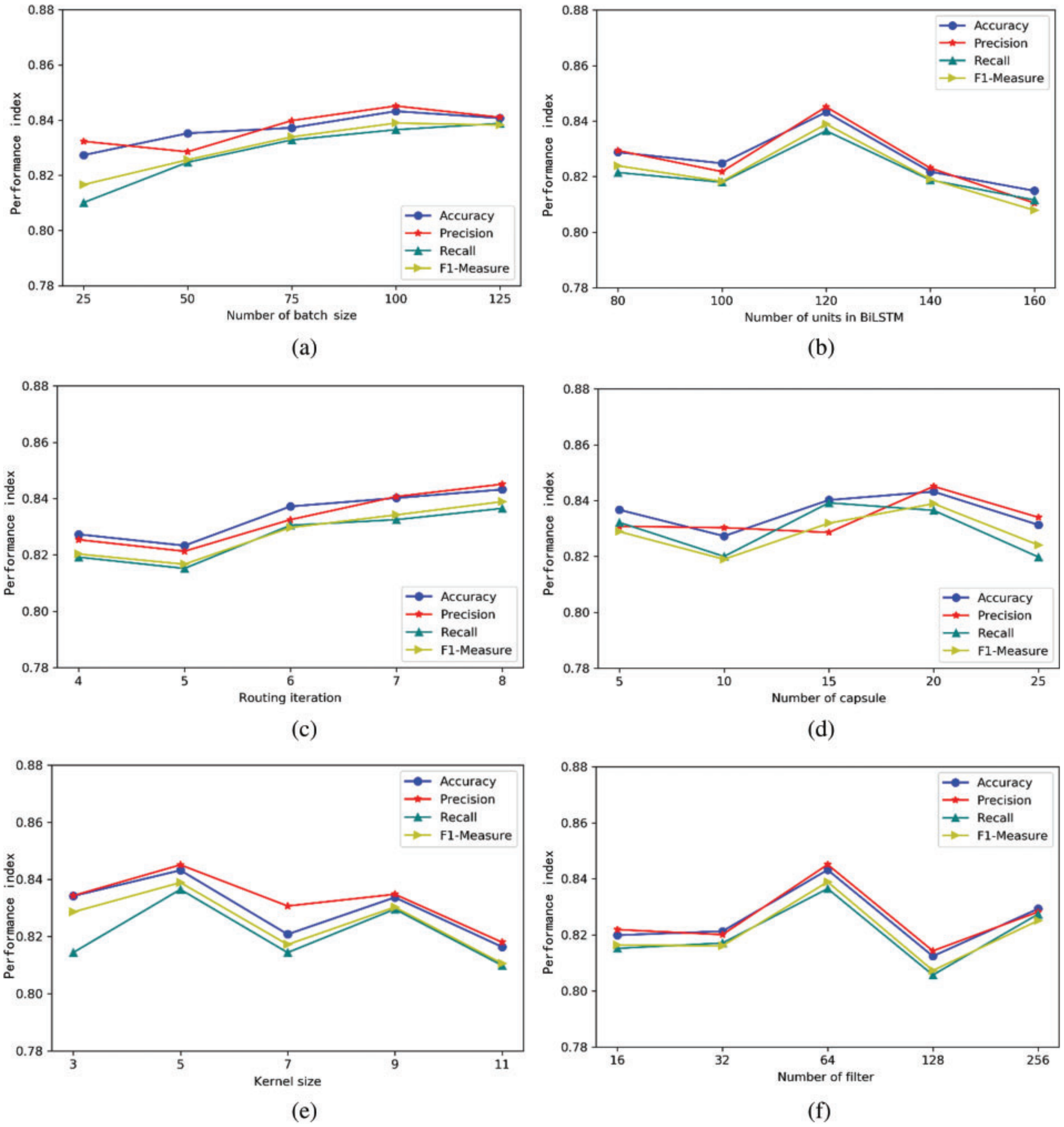
To verify the effectiveness of the institutional attribute mining method proposed in this paper, we conducted simulation experiments based on a medical triage department text resource dataset [31], aiming to mine the institutional attribute information corresponding to unstructured text resources. This dataset covers institutional attribute information from 15 hospital departments, comprising a total of 10,072 unstructured text resources. Detailed descriptions of the dataset are provided in Table 2. By randomly splitting the dataset, we obtained a training dataset consisting of 8057 (80%) data entries and a testing dataset consisting of 2015 (20%) data entries. This dataset is representative for validating our method. This is because the selected dataset encompasses a wide variety of medical text data resources, which are representative in terms of linguistic style, terminology usage, and content complexity. Additionally, the scale of the dataset is sufficient to evaluate experimental performance, allowing it to capture both common patterns and anomalies within the medical field. The distribution of samples across different categories in the dataset is well-balanced, avoiding the issue of class imbalance and ensuring the quality and reliability of the data. For various real-world unstructured text resources outside the medical domain, the model's processing approach is exactly the same. During the experiment, the word vector dimension used in the input layer was 3072, employing the BERT pre-trained language model. The categorical cross-entropy was used as the loss function, and the Adam optimizer was utilized as the model training optimizer, with the optimizer's learning rate set to 1×10^{-3} to minimize the total Loss value during training. The experimental software and hardware environment is as follows: the operating system is Win10 64-bit, CPU is Intel(R) Core(TM) i7-4710MQ @ 2.5 GHz, GPU is GeForce GTX 850M, memory size is 16 GB, Python version is 3.6, Tensorflow version is 1.14.0, and Keras version is 2.1.3. We designed two experiments to evaluate and compare the performance of institutional attribute mining: one compares the performance of different hyperparameters, and the other compares the performance with benchmark methods.

Table 2: Data set description

Institutional attribute	Number of resources	Institutional attribute	Number of resources	Institutional attribute	Number of resources
Pediatrics department	1231	Gastrology department	882	Neurology department	863
Dermatology department	750	Orthopedics department	736	Obstetrics and gynecology department	730
Cardiovascular medicine department	653	Breast Surgery department	631	Ophthalmology department	625
Stomatology department	607	Endocrinology department	539	Hepatological surgery department	491
Urinary surgery department	477	Rheumatology and immunology department	436	Hematology department	421

4.2 Performance Comparison of Models with Different Hyperparameters

The purpose of the performance comparison experiment of models with different hyperparameters is to verify the effectiveness of the proposed method for mining institutional attributes, specifically whether it can correctly identify the institutional attribute information corresponding to unstructured text resources, and to examine the impact of different hyperparameters on the model's learning performance. The experimental results are shown in Fig. 6a–f. The experiments separately validated and compared the effects of batch size (the amount of data input per training session), the number of units in the BiLSTM structure, the number of routings in the Capsule Network structure, the number of capsules in the Capsule Network structure, the kernel size in the Convolutional structure, and the number of filters in the Convolutional structure on model performance. We evaluated the experimental performance from four dimensions: Accuracy, Precision, Recall, and F1-Measure.



(a) (b) (c) (d) (e) (f)

Figure 6: Performance comparison of different hyperparameters. (a) Different batch_sizes; (b) Different neuron units in BiLSTM structure; (c) Different routes in capsule network; (d) Different capsules in capsule network; (e) Different convolution kernel sizes; (f) Different number of filters

Fig. 6a shows the experimental results of the effect of batch size on model performance. From the experimental results, it can be seen that as the batch size increases, the trend of changes in the model's performance metrics steadily increases. This is because, with an increase in the batch size, the model can train more effectively based on a larger amount of data. However, when the batch size grows to a certain scale, it can lead to the training process reaching overfitting too quickly, thus affecting further improvements in training performance. Through experimentation, it was found that a batch size of 100 yields relatively optimal overall model performance. Fig. 6b presents the experimental results of the impact of the number of units in the BiLSTM structure on model performance. The results show that when the number of units is 120, the model exhibits relatively optimal overall performance, capable of sufficiently mining semantic information from the context of the text. Thus, setting excessively high numbers of units is unnecessary, as it can reduce the efficiency of model training. Additionally, during the experiment, it was observed that the number of units in the BiLSTM affects the speed of convergence during training; as the number of units increases, the training converges more slowly. Therefore, selecting excessively high numbers of units is not necessary. Fig. 6c illustrates the experimental results of the impact of the number of routings in the Capsule Network structure on model performance. The results indicate that when the number of routings is 8, the model demonstrates relatively optimal overall performance. Under these conditions, the dynamic routing mechanism can achieve better feature selection among different features, balancing the influence of various features on the model and selecting superior features for mining institutional attributes. Fig. 6d shows the experimental results of the impact of the number of capsules in the Capsule Network structure on model performance. When the number of capsules is 20, the model performs best. Fig. 6e presents the experimental results of the impact of the kernel size in the convolutional structure before the GatedConv unit on model performance. The effect of kernel size on model performance is related to the representation form of word vectors in text resources. Too short a kernel size will fail to effectively mine the spatial relationships between word vectors with sequential associations, whereas too long a kernel size can lead to the loss of spatial relationships between sequence word vectors, causing the attention to be dispersed and affecting experimental performance. Experimental findings suggest that a kernel size of 5 can better balance the association between words, leading to the best model performance. Fig. 6f illustrates the experimental results of the impact of the number of filters in the convolutional structure on model performance. When the number of filters is 64, the model performs best. An insufficient number of filters can lead to underfitting of the institutional attribute mining model, while an excessive number of filters can cause overfitting, impacting further enhancements in system performance.

In order to verify the independent contribution of each module in the hybrid capsule network model constructed in this paper to the final effect, we conducted a combined ablation experiment of the BiLSTM module, the Attention module and the Capsule module in the hybrid model. The independent contribution of each module to the attribute mining performance of the final mechanism was verified through ablation experiments to quantify the degree of improvement of the hybrid model on the final performance. The results of the ablation experiment are shown in Table 3. The introduction of the attention mechanism based on the BiLSTM module can enhance the attention to important features, thus improving the model performance. The single capsule network and BiLSTM+Attention network have similar institution attribute mining performance. Through the parallel architecture of the hybrid model, the physical vision of the data feature information concerned by the two network structures is different. Different modules can focus on the feature processing of the parts they are good at processing, and fuse the higher-order features obtained further, so that the fused features can pay attention to the local features and global features of the text data at the same time, and carry out a more comprehensive description of the features of the text data, so as to further improve the mining performance of the institution attribute.

Table 3: Results of ablation experiments

ID	Choice			Performance index			
	BiLSTM module	Attention module	Capsule module	Accurary	Precision	Recall	F1-Measure
1	✓			0.7571	0.7611	0.7561	0.7571
2	✓	✓		0.7886	0.7815	0.7876	0.7826
3			✓	0.7864	0.7839	0.7864	0.7851
4	✓		✓	0.8212	0.8204	0.8199	0.8201
5	✓	✓	✓	0.8432	0.8451	0.8365	0.8389

4.3 Performance Comparison with Baseline Methods

To compare the performance differences of various neural network models in the task of mining institutional attributes, we selected seven commonly used neural network models as baseline comparison models. The description information of the baseline comparison models is shown in Table 4.

Table 4: Baseline comparison model

Number	Model name	Model structure	Hyperparameter setting

1	CNN model	First add CNN network structure, and then add a full connection layer.	filters = 128,
			kernel_size = 5,
			dense_units = 64

2	CNN_LSTM model	First add CNN network, then add LSTM network.	filters = 32,
			kernel_size = 3,
			lstm_units = 100

3	BiLSTM model	Bidirectional LSTM network architecture.	lstm_units = 128
			return_sequences = False

4	ATT+BiLSTM model	First add attention structure, then add bidirectional BiLSTM network.	lstm_units = 128
			return_sequences = False

5	CNN_GRU model	First add CNN network, then add GRU network.	filters = 32,
			kernel_size = 3,
			gru_units = 100

6	BiGRU model	Bidirectional GRU network struc-ture.	gru_units = 100
			return_sequences = False

7	ATT+BiGRU model	First Add attention structure, then add two-way GRU network.	gru_units = 100
			return_sequences = False

From the experimental results shown in Fig. 7a–d, it can be seen that compared to the performance of other baseline methods, the method proposed in this paper performs optimally in all four evaluation metrics. For the problem of institutional attribute mining, a single network architecture may experience performance degradation as the length of text resources increases due to issues such as vanishing gradients. In contrast, a hybrid capsule network that combines attention mechanisms, BiLSTM, and capsule networks leverages the strengths of each component to form a powerful and flexible framework for institutional attribute mining. This framework not only effectively captures the sequential information and structural features of text but also possesses excellent adaptability and reasoning capabilities, making it suitable for a wide range of complex text attribute mining tasks. Firstly, the hybrid capsule network integrates local and global feature information. It has been specifically optimized for the domain of institutional attribute mining, enhancing representation learning and semantic understanding while reducing boundary errors in attribute extraction. The BiLSTM excels at capturing local context information within the text, whereas the capsule network can grasp the overall structure of the text and the relationships between its components. By combining these two, the model can generate richer and more comprehensive text representations, thereby improving the accuracy of attribute mining. The attention mechanism allows the model to further highlight key content based on the sequential information provided by the BiLSTM. The structured representation of the hybrid capsule network aids in understanding deeper semantics of the text, such as themes, arguments, and argument structures. Secondly, the hybrid capsule network exhibits dynamic adaptability. The combination of the attention mechanism and the dynamic routing of the capsule network enables the model to flexibly adjust its internal states according to the specific requirements of different tasks. For instance, when dealing with various types of text resources, the model can automatically focus on the most appropriate features, enhancing its flexibility and generalization ability. Lastly, the parallel network structure of the hybrid capsule network strengthens its reasoning capability. The layered structure of the capsule network allows the model to construct higher-level abstract concepts, which assist in performing complex reasoning. Combined with the attention mechanism, the model can make more intelligent decisions about which parts should be emphasized and which can be disregarded, thus improving the quality of decision-making. For text resources with multiple layers of meaning or complex logical structures, the combined model can achieve more precise attribute mining by capturing features at multiple levels. Therefore, this approach can achieve better overall results in institutional attribute mining. There is a comprehensive performance improvement ranging from 2.06% to 8.18% in the F1-Measure metric compared to other methods, making it better suited to meet the application needs of institutional attribute mining for access control.

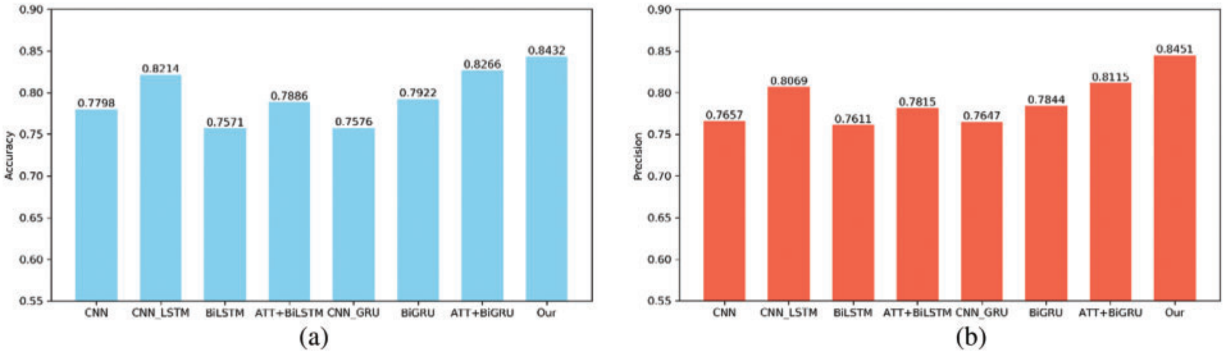


Figure 7: (Continued)

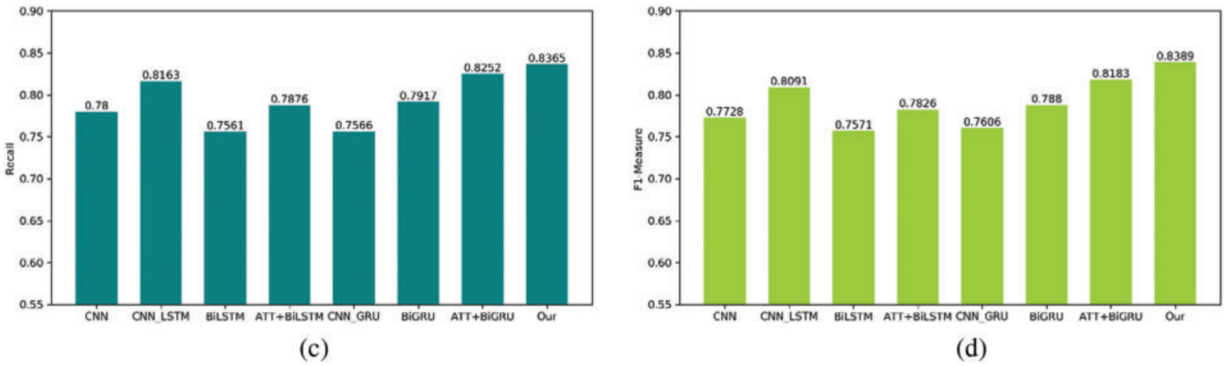


Figure 7: Performance comparisons with different benchmark methods. (a) Accuracy comparison of different models; (b) Precision comparison of different models; (c) Recall comparison of different models; (d) F1-Measurecomparison of different models

We compared the efficiency of attribute mining achieved by different benchmark methods. According to the experimental results presented in Table 5, our proposed method takes the longest time for attribute mining of a single text resource compared to other mining methods. This is primarily due to the more complex network architecture and the larger number of model parameters. However, under the experimental conditions, the average time to process a single text resource is approximately 31.2 ms, which is sufficient to meet the real-time requirements for attribute mining.

Table 5: Baseline comparison model

Methods	CNN	CNN_LSTM	BiLSTM	ATT+BiLSTM	CNN_GRU	BiGRU	ATT+BiGRU	Our
Time (ms)	12.9	17.6	20.2	23.1	24.3	19.7	22.7	31.2

5 Conclusion

To address the challenge of automated institutional attribute mining in attribute-based access control mechanisms, this paper proposes an access control institutional attribute mining technique based on a hybrid capsule network. Adopting the concept of transfer learning, this technique integrates Long Short-Term Memory (LSTM) networks, attention mechanisms, capsule networks, and dynamic routing mechanisms to design and implement a novel end-to-end parallel hybrid network structure. This structure can incorporate both global and local textual feature information simultaneously, providing a new solution pathway for generating institutional attributes for access control systems. Experimental results demonstrate that, compared to baseline algorithms, our method achieves better performance on the test dataset, reducing mining errors and enhancing the efficiency of security administrators' attribute management. In future work, we will further strive to improve the performance of institutional attribute mining and explore high-performance training methods under conditions of small sample datasets.

Acknowledgement: We thank all the members who have contributed to this work with us.

Funding Statement: This work is supported by National Natural Science Foundation of China (No. 62102449).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Aodi Liu, Xuehui Du; data collection: Aodi Liu, Xiangyu Wu; analysis and interpretation of results: Aodi Liu, Na Wang,

Xiangyu Wu; draft manuscript preparation: Aodi Liu, Na Wang, Xuehui Du; validation: Aodi Liu, Xiangyu Wu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available in Baidu Cloud at https://pan.baidu.com/s/1UDqILTUCAGacMPZ3h-VyuA?pwd=z12b, accessed on 01 January 2025.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Liu L, Li J, Lv J, Wang J, Zhao S, Lu Q. Privacy-preserving and secure industrial big data analytics: a survey and the research framework. IEEE Internet Things J. 2024;11(11):18976–99. doi:10.1109/JIOT.2024.3353727.
2. Zhang R, Liu G, Kang H, Wang Q, Wan B, Luo N. Anonymity in attribute-based access control: framework and metric. IEEE Trans Dependable Secur Comput. 2024;21(1):463–75. doi:10.1109/TDSC.2023.3261309.
3. Mishra AK, Wazid M, Singh DP, Das AK, Roy S, Shetty S. ACKS-IA: an access control and key agreement scheme for securing Industry 4.0 applications. IEEE Trans Netw Sci Eng. 2024;11(1):254–69. doi:10.1109/TNSE.2023.3296329.
4. Arshad H, Johansen C, Owe O. Semantic attribute-based access control: a review on current status and future perspectives. J Syst Archit. 2022;129(6):102625. doi:10.1016/j.sysarc.2022.102625.
5. Xiao MY, Li HB, Huang Q, Yu S, Susilo W. Attribute-based hierarchical access control with extendable policy. IEEE Trans Inf Forensics Secur. 2022;17:1868–83. doi:10.1109/TIFS.2022.3173412.
6. Gupta E, Sural S, Vaidya J, Atluri V. Enabling attribute-based access control in NoSQL databases. IEEE Trans Emerg Top Comput. 2023;11(1):208–23. doi:10.1109/TETC.2022.3193577.
7. Mohamed A, Najafabadi MK, Wah YB, Zaman EAK, Maskat R. The state of the art and taxonomy of big data analytics: view from new big data framework. Artif Intell Rev. 2020;53(2):989–1037. doi:10.1007/s10462-019-09685-9.
8. The Evolution of Data to Life-Critical Don't Focus on Big Data; Focus on the Data That's Big; [cited 2024 Oct 10]. Available from: https://www.import.io/wp-content/uploads/2017/04/Seagate-WP-DataAge2025-March-2017.pdf.
9. Liu A, Du X, Wang N. Unstructured text resource access control attribute mining technology based on convolutional neural network. IEEE Access. 2019;7:43031–41. doi:10.1109/ACCESS.2019.2907815.
10. Alohaly M, Takabi H, Blanco E. Automated extraction of attributes from natural language attribute-based access control (ABAC) policies. Cybersecurity. 2019;2(1):2. doi:10.1186/s42400-018-0019-2.
11. Alohaly M, Takabi H, Blanco E. A deep learning approach for extracting attributes of ABAC policies. In: Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies; 2018 Jun 13–15; Indianapolis, IN, USA; p. 137–48.
12. Chen LC, Chang KH. An entropy-based corpus method for improving keyword extraction: an example of sustainability corpus. Eng Appl Artif Intell. 2024;133(4):108049. doi:10.1016/j.engappai.2024.108049.
13. Zhang L, Li Y, Li Q. A graph-based keyword extraction method for academic literature knowledge graph construction. Mathematics. 2024;12(9):1349. doi:10.3390/math12091349.
14. Du H, Thudumu S, Giardina A, Vasa R, Mouzakis K, Jiang L. Contextual topic discovery using unsupervised keyphrase extraction and hierarchical semantic graph model. J Big Data. 2023;10(1):156. doi:10.1186/s40537-023-00833-1.
15. Abasi AK, Khader AT, Al-Betar MA, Naim S, Makhadmech SN. An ensemble topic extraction approach based on optimization clusters using hybrid multi-verse optimizer for scientific publications. J Ambient Intell Humaniz Comput. 2021;12(2):2765–801. doi:10.1007/s12652-020-02439-4.
16. Gadekar H, Bugalia N. Automatic classification of construction safety reports using semi-supervised YAKE-Guided LDA approach. Adv Eng Inform. 2023;56:101929. doi:10.1016/j.aei.2023.101929.
17. Waqas M, Anjum N, Afzal MT. A hybrid strategy to extract metadata from scholarly articles by utilizing support vector machine and heuristics. Scientometrics. 2023;128(8):4349–82. doi:10.1007/s11192-023-04774-7.

18. Wu D, Cheng P, Zheng Y. Multistage mixed-attention unsupervised keyword extraction for summary generation. Appl Sci. 2024;14(6):2435. doi:10.3390/app14062435.
19. Zhang S, Shen Y, Tan Z, Wu Y, Lu W. De-bias for generative extraction in unified NER task. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics; 2022 May 22–27; Dublin, Ireland; p. 808–18.
20. Qiu Q, Tian M, Huang Z, Xie Z, Ma K, Tao L. Chinese engineering geological named entity recognition by fusing multi-features and data enhancement using deep learning. Expert Syst Appl. 2024;238(1):121925. doi:10.1016/j.eswa.2023.121925.
21. Kim Y. Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014 Oct 25–29; Doha, Qatar.
22. Peng H, Li J, Wang S, Wang L, Gong Q, Yang R. Hierarchical taxonomy-aware and attentional graph capsule RCNNs for large-scale multi-label text classification. IEEE Trans Knowl Data Eng. 2021;33(6):2505–19. doi:10.1109/TKDE.2019.2959991.
23. Lu XS, Zhou M, Wu K. A novel fuzzy logic-based text classification method for tracking rare events on twitter. IEEE Trans Syst Man Cybern: Syst. 2021;51(7):4324–33. doi:10.1109/TSMC.2019.2932436.
24. Cheng Y, Zou H, Sun H, Chen H, Cai Y, Li M. HSAN-capsule: a novel text classification model. Neurocomputing. 2022;489:521–33. doi:10.1016/j.neucom.2021.12.064.
25. Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M, Gao JF. Deep learning-based text classification: a comprehensive review. ACM Comput Surv. 2021;54(3):62. doi:10.1145/343972.
26. Mahmoud HAH, Hafez AM, Alabdulkreem E. Language-independent text tokenization using unsupervised deep learning. Intell Autom Soft Comput. 2023;35(1):321–34. doi:10.32604/iasc.2023.026235.
27. Zhang J, Xu QH, Guo LL, Ding L, Ding SF. A novel capsule network based on deep routing and residual learning. Soft Comput. 2023;27(12):7895–906. doi:10.1007/s00500-023-08018-x.
28. He P, Zhou Y, Duan SK, Hu XF. Memristive residual CapsNet: a hardware friendly multi-level capsule network. Neurocomputing. 2022;496(10):1–10. doi:10.1016/j.neucom.2022.04.088.
29. Kenton JDM-WC, Toutanova LK. Bert: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT; 2019 Jun 2–7; Minneapolis, MN, USA. p. 2.
30. Sun Y, Qiu HP, Zheng Y, Zhang CR, Hao C. Knowledge enhancement for pre-trained language models: a survey. J Chin Inf Process. 2021;35(7):10–29 (In Chinese).
31. Medical referral text resource dataset [Online] [cited 2025 Jan 01]. Available from: https://pan.baidu.com/s/1UDqILTUCAGacMPZ3h-VyuA?pwd=z12b.