Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.055167

ARTICLE

Automatic Generation of Attribute-Based Access Control Policies from Natural Language Documents

Fangfang Shan[1,2,*], Zhenyu Wang[1,2], Mengyao Liu[1,2] and Menghan Zhang[1,2]

[1]College of Computer, Zhongyuan University of Technology, Zhengzhou, 450007, China

[2]Henan Key Laboratory of Cyberspace Situation Awareness, Zhengzhou, 450001, China

*Corresponding Author: Fangfang Shan. Email: 6129@zut.edu.cn

Received: 19 June 2024 Accepted: 02 August 2024 Published: 12 September 2024

ABSTRACT

In response to the challenges of generating Attribute-Based Access Control (ABAC) policies, this paper proposes a deep learning-based method to automatically generate ABAC policies from natural language documents. This method is aimed at organizations such as companies and schools that are transitioning from traditional access control models to the ABAC model. The manual retrieval and analysis involved in this transition are inefficient, prone to errors, and costly. Most organizations have high-level specifications defined for security policies that include a set of access control policies, which often exist in the form of natural language documents. Utilizing this rich source of information, our method effectively identifies and extracts the necessary attributes and rules for access control from natural language documents, thereby constructing and optimizing access control policies. This work transforms the problem of policy automation generation into two tasks: extraction of access control statements and mining of access control attributes. First, the Chat General Language Model (ChatGLM) is employed to extract access control-related statements from a wide range of natural language documents by constructing unique prompts and leveraging the model's In-Context Learning to contextualize the statements. Then, the Iterated Dilated-Convolutions-Conditional Random Field (ID-CNN-CRF) model is used to annotate access control attributes within these extracted statements, including subject attributes, object attributes, and action attributes, thus reassembling new access control policies. Experimental results show that our method, compared to baseline methods, achieved the highest F1 score of 0.961, confirming the model's effectiveness and accuracy.

KEYWORDS

Access control; policy generation; natural language; deep learning

1 Introduction

In this digital age full of uncertainties, information has become one of the most crucial assets. Information security is a key element in building the foundation of trust in modern society. Whether it's e-commerce platforms, banks, or health service providers, users expect their data to be handled and protected with care. The increasing news of privacy breaches, data misuse, and security lapses serves as a reminder of the risks associated with inadequate information security measures. Individuals and businesses suffer when personal information becomes the target of thieves or is accessed or disclosed

without proper authorization. Every click, message, transaction, and service interaction could involve the exchange of sensitive information. For individuals, enterprises, and even national institutions that rely on this information, protecting information security has become an urgent task. The significance of information security is not only limited to technical defenses; it also represents trust, responsibility, and durability.

One of the crucial methods for protecting information security is access control [1], whose task is to ensure that information resources are not used or accessed illegally. Conflict detection and resolution primarily address the problem of inconsistent security policies across different information systems. An access control system first verifies the identity of the user or system requesting access to the resources, and then determines their rights to access these resources based on predefined security policies. This process involves two phases: authentication and authorization, ensuring that only the appropriate individuals or systems access the right resources at the right time. By limiting data access, access control helps protect the sensitive information of user organizations from unauthorized access, while reducing the possibility of data leaks, theft, or damage. It can also log access data and user activity, providing logs for security monitoring. These logs are crucial for the investigation of security incidents and compliance auditing since they can help reconstruct the timeline of events, determine responsibilities, and take appropriate remedial actions. ABAC (Attribute-Based Access Control) [2] is an access control model that defines a decision-making framework allowing systems to grant access decisions based on the policies defined by developers or administrators, considering the attributes of the requester, the resource, the environment, and other possible contextual properties. In ABAC [3], the granting of permissions is based not only on the identity of the user but also on a set of attributes, including those of the user, the resource, as well as other relevant environmental attributes. Its widespread use stems from two core advantages: a high degree of flexibility and granularity. Compared to traditional RBAC (Role-Based Access Control) [4], ABAC can handle a greater variety of variables and conditions [5], thereby offering more precise and dynamic access management. For example, an ABAC system may permit access for requests originating from a specific geographic location during certain time frames, rather than solely based on the user's role. This enables authorization decisions to be made based on a rich context. It allows for nuanced control over access requests to a given resource, meeting security requirements while ensuring users' access needs are met.

Access control policies [6] refer to a set of rules or guidelines that define how, when, and under what conditions users (or user groups), programs, or other digital identities are allowed to access specific information technology resources. These policies are a key part of ensuring the security of information systems, used to prevent unauthorized access and protect sensitive data from being leaked or damaged. Access control policies can be categorized into different types [7] based on their implementation. In Role-Based Access Control, permissions are assigned to roles based on the user's role (e.g., manager, employee, etc.), and users obtain permissions corresponding to their roles. In Attribute-Based Access Control, permissions are dynamically granted based on attributes (e.g., the user's location, time, etc.), which can be attributes of the user, the resource, or environmental attributes. In Mandatory Access Control, mandatory security policies set by the operating system or security administrators typically rely on multi-level security labels, and users cannot modify these settings.

Currently, technologies for automatic generation of access control policies mainly fall into two categories: those aimed at system access logs and natural language documents. The techniques focusing on system access logs primarily analyze user-permission relationships within access logs. This includes determining which users, programs, or devices should have access rights, and what level of access control should be applied to resources, followed by the formulation of new relevant access control policies.

In many organizations and institutions' information systems, there is a widespread presence of documents compiled in natural language, such as project requirement documents, user manuals, and operation guides. These documents not only detail the system functionalities and operational procedures but also often contain inherent access control principles and policies [8]. Extracting access control policies from natural language documents is a complex task, mainly facing the following challenges: 1) Natural language often contains ambiguous, unclear, and polysemous expressions, making it difficult to parse and understand the intent of the document. 2) The syntactic structures and styles of natural language documents can vary greatly. The same policy can be expressed in many different ways, requiring the extraction process to handle various complex syntactic structures and linguistic styles. 3) Different industries or organizations may use specific terminology and expertise, necessitating background knowledge in the relevant field to extract policies from documents. 4) Access control policies often depend on specific contextual environments, and the extraction process needs to accurately understand and apply this contextual information to ensure the correctness and completeness of the policies. Therefore, researching how to effectively and automatically extract these implicit access control policies from natural language documents and convert them into actionable ABAC policies is crucial for the development of attribute-based access control technology.

This article aims to propose a deep learning-based technique for the automatic generation of attribute-based access control policies. The goal of this technique is to automatically extract and generate attribute-based access control policies from natural language documents. It intelligently identifies and refines the necessary attributes and rules for access control from unstructured texts such as project requirements and user manuals, thereby constructing and optimizing access control policies. This enhances the speed and intelligence level of access control policy formulation, providing more efficient and automated support for access control policies. This paper's contributions are as follows:

(1) This work introduced a novel approach based on deep learning to autonomously generate attribute-based access control policies from natural language documents. We delineate the policy generation challenge into two core tasks: the extraction of access control statements and the mining of attributes within those statements;

(2) This work employed the ChatGLM pre-trained model to undertake the task of extracting access control statements. The task of mining statement attributes is transformed into an attribute annotation challenge, where we utilize the ID-CNN-CRF model for the sequence labeling of attributes. Experimental results demonstrate the model's excellent performance;

(3) This work designed the OBIE attribute annotation method that allows for more detailed annotation of subject, object, and action attributes within access control statements. This method enhances the model's performance optimization.

The rest of the paper is organized as follows: Section 2 begins with an overview of previous literature and information about related work. The access control policy generation framework and its components are presented in Section 3. Section 4 presents the experiments and results and discussion. Finally, Section 5 presents conclusions.

2 Related Work

Currently, scholars both domestically and internationally have conducted a series of research in the field of automatic generation of access control policies. These mainly fall into two types: the first type targets the existing access control information, namely the access logs in the system. Cotrini et al. [9]

proposed a method for extracting access control policies from sparse logs, called Rhapsody, which uses association rule mining to effectively process sparse logs and employs a new reliability measure to avoid mining rules that over-privilege. In addition, they introduced a novel validation method "generic cross-validation", which, compared with standard cross-validation, uses requests that do not appear in the logs to evaluate the mined rules, thus improving the quality of the rules; Das et al. [10] developed a new ABAC policy mining algorithm based on a greedy heuristic, which considers environmental attributes and their related values, and uses the Gini impurity to form rules, aimed at minimizing the number of rules in the generated policy. Iyer et al. [11] introduced a novel approach for mining ABAC policies, where previous studies mainly focused on mining only positive authorization rules, this method can process both positive and negative authorization rules and also requires less time cost than previous methods; Shang et al. [12] used decision trees to segment and encode log information to generate a matrix representation characterizing the logs. Hierarchical clustering is then used to cluster similar logs and extract attribute relationships to construct the ABAC policy set; Bamberger et al. [13] proposed an access control policy generation algorithm that takes as input a set of access request logs as well as attributes of entities in the system and optionally existing policies to generate structured attribute-based access control policies; Sanders et al. [14] presented a rule-mining based method that can generate ABAC policies in larger and more complex audit logs, defining the ABAC permission error minimization problem, which is used to balance the issues of insufficient and excessive permissions in security policies; Mocanu et al. [15] employed a restricted Boltzmann machine (RBM) trained on logs to extract policy rules. However, this study only presented preliminary results for the first phase in policy space, and the final phase of the algorithm has not yet been realized; Xu et al. [16] proposed an algorithm for mining ABAC policies from operational logs and attribute data, taking into account noise in the logs. Rules with quality measures below a threshold are considered noise and are deleted to improve the quality of the mined policies; Karimi et al. [17] presented a method to automatically learn access control policy rules from system access logs. This method uses the unsupervised learning technique of K-modes clustering to detect access log patterns, preliminarily extracting access control policy authorization rules and employs rule pruning and policy refinement to optimize the policy quality and conciseness. Rule pruning uses the Jaccard similarity index to eliminate rules that contribute little or redundant to policy quality improvement, thus compressing the policy size. However, the method has issues with the stability of the policy generation quality and the difficulty in setting appropriate clustering values;

The second approach does not require existing access control information and targets natural language specification documents. Ammar et al. [18] addressed the problem of existing website privacy policies being obscure and verbose by proposing an automatic text classification method. They used a logistic regression model to analyze privacy policy texts and search for significant features, mapping privacy policy documents to classification labels and incorporating self-training semi-supervised techniques to improve classification accuracy for this task; Xiao et al. [19] introduced an automated approach named Text2Policy, which combines syntactic and semantic analysis to extract model instances and generate formal specifications. It matches against four predefined policy semantic patterns, capable of extracting RBAC policies from documents containing access control policies, and automatically deriving access control requests that can verify against specified or extracted ACPs to detect inconsistencies, from the operation steps extracted; Zhu et al. [20] proposed a hybrid neural network model that enables dynamic mining of content attributes that accurately and completely express the semantics of unstructured text in a massive heterogeneous data environment; Xia et al. [21] proposed to present a comprehensive automated approach for extracting ABAC policies from natural language documents in healthcare systems, which fully utilizes BERT and Semantic

Role Labeling (SRL) in ACP utterance recognition and rule extraction, where the BERT model is used to classify the decision results of ACP utterances and rules, and to generate semantic role labels and embedded attribute values. And the SRL representation is utilized to enhance the BERT-based ACP utterance recognition; Slankas et al. [22] suggested a machine-learning-based method to extract implicit and explicit access control policies from natural language documents. They first input the entire text, breaking it down into types such as titles, list beginnings, list items, and ordinary sentences. They then use machine learning algorithms to classify sentences as access control or non-access control, followed by the use of relation extraction methods to identify and extract access control elements, and finally, they inspect the coverage and conflicts of the extracted access control policies. Narouei et al. [23] proposed a top-down policy engineering framework aimed at automatically extracting policies from natural language documents. They trained a deep recurrent neural network that uses pre-trained word embeddings to identify sentences containing access control policy content; subsequently, Narouei et al. [24] introduced a framework that utilizes Semantic Role Labeling (SRL) to extract access control policies from unrestricted natural language documents. SRL extracts contextual information and conditions of the environment, facilitating the precise definition of access control policies. This approach further improves the performance of SRL in extracting access control policies by utilizing domain adaptation and semi-supervised learning techniques; Alohaly et al. [25] developed an innovative framework capable of extracting attributes for attribute-based access control policies from natural language documents. They designed this process as a problem of relation extraction, systematically linking subjects and objects and the attributes that modify these elements, transforming attribute-based access control policies from unstructured natural language descriptions into a format that machines can parse and execute.

There are also some recent advances on natural language processing techniques, such as research on the security of LLMs (Large Language Models). Zhao et al. [26] proposed a new clean label backdoor attack method Cbat with text style, which does not require external triggers and the poisoned samples can be labeled correctly, and also proposed an algorithm CbatD to defend the backdoor attack, which effectively eliminates the poisoned samples by finding the minimum training loss and calculating the feature correlation; You et al. [27] proposed an attack method, LLMBkd, which improves the effectiveness of textual backdoor attacks by automatically inserting various style-based triggers into text using a language model. The defense method REACT, a baseline defense that mitigates backdoor attacks by antidote training examples, is also proposed; Shao et al. [28] developed two CTF-solving workflows, human-in-the-loop (HITL) and fully-automated, which comprehensively evaluated the ability of LLM-solving in solving real-world CTF challenges, and provided a reference for applying LLM in cybersecurity education.

3 Access Control Policy Generation

3.1 Access Control Policy Generation Framework

In the area of text data feature extraction, significant advancements have been made in machine learning, deep learning, and natural language processing technologies. This study proposes a deep learning-based approach for automatically extracting Attribute-Based Access Control (ABAC) policies from project specification documents described in natural language. This method is aimed at automating and intelligently generating the system's access control policies, significantly reducing the time required to establish such policies. The paper starts by extracting statements related to access control from natural language documents, followed by mining the attributes within these statements, including subject attributes, object attributes, and action attributes, and then formulates new, readable,

and effective access control policies. The overall framework for access control policy generation is depicted in Fig. 1.

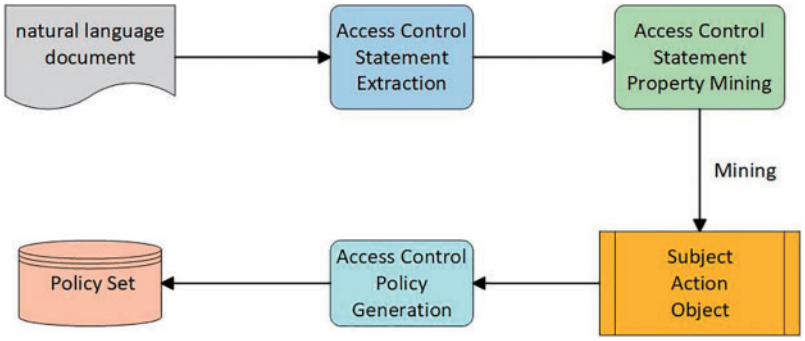

Figure 1: Policy generation framework

3.2 Extraction of Access Control Policy Statements

3.2.1 Access Control Policy Statement Extraction Process

ChatGLM(Chat General Language Model) [29] is a pioneering pre-trained language model, distinguished primarily for its open-source nature and optimizations tailored for Chinese. Remarkably, it facilitates the deployment of a lightweight int4 service directly on personal hardware, significantly lowering deployment expenses. To cater to developers aiming to adapt the model for bespoke application contexts, ChatGLM introduces an adept model fine-tuning methodology grounded on P-Tuning v2. This approach is particularly effective for tasks like information extraction, with the following instance serving as an illustrative example of extracting data from datasets.

A user posted the following information: "I am an artist who loves to travel, to see the world's beautiful scenery, and to engage in everyday conversations. I am someone who values authenticity and always finds topics to discuss, even if it's just casual chatter. Born on 30 October, 1999, I am a Scorpio and currently reside in Beijing, attending the Beijing Art Academy. I joined Weibo on 12 March, 2016."

Using the ChatGLM large model, information extracted from the aforementioned example is as follows:

{"Username": "Not provided",

"Bio": "Enjoys traveling, taking in the beautiful scenes of the world, and engaging in daily conversations. It's often about casual and authentic talks; just lending an ear is good enough. There's always something to discuss for someone who values truth."

"Birthday": "1999-10-30",

"Additional Information":

{"Zodiac Sign": "Scorpio",

"City": "Beijing",

"Alma Mater": "Beijing Film Academy",

"Weibo Joining Date": "2016-03-12"}}.

This paper proposes to utilize the adapted ChatGLM model to identify and extract access control statements from datasets.

To successfully enable ChatGLM to extract access control statements, we need to design a unique prompt. First, classify the corpus to determine its conceptual hierarchy, such as identifying entities like people, actions, etc. This will help the model better understand the characteristics and context of the corpus. Next, we need to define the content and requirements of the access control statements. These statements may involve aspects such as permission management, authentication, data access restrictions, etc. By clearly defining the access control statements, we can guide ChatGLM to be more accurate and targeted during the extraction process.

In the process of building the prompt, listing some correct examples as context for In-Context Learning is beneficial. These examples may include common access control statements, along with variants and extensions in specific scenarios. By introducing these examples, the aim is to enhance ChatGLM's understanding of different contexts and usages, thereby improving its performance in the task of access control statement extraction.

Finally, providing natural language specifications and documents allows ChatGLM to extract access control statements from them. These documents could encompass best practices in access control, security policy regulations, and guidelines for permission management, among others. A flowchart for the access control statement extraction process is depicted in Fig. 2.

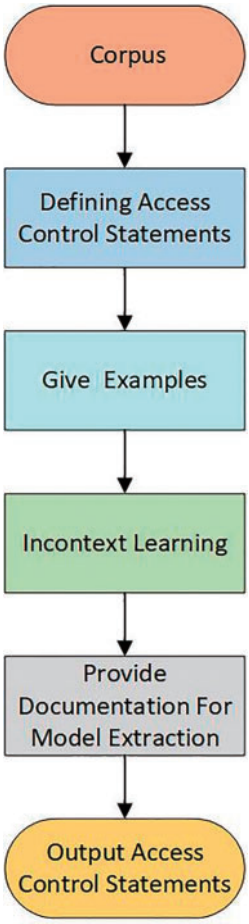

Figure 2: Access control statement extraction process flowchart

3.2.2 P-Tuning v2 Method

In this paper, we use the P-Tuning v2 [30] method to fine-tune the ChatGLM model, which utilizes Deep Prefix Tuning with improvements to Prompt Tuning and P-Tuning as a cross-scale and NLU (Natural Language Understanding) generalized solution for the task. There are two main problems with previous approaches such as Prompt Tuning and P-Tuning: 1) Lack of model parameter size and task generalization, for those smaller models (from 100M to 1B), the performance of Prompt Optimization and full fine-tuning varies considerably. 2) Lack of Depth Prompt Optimization, due to the limitations of the sequence lengths, the number of tunable parameters is limited.

The P-Tuning v2 method applies Prefix-tuning to in NLU tasks, using a multi-task learning optimization, pre-training based on Prompt from a multi-task dataset and then adapted downstream tasks. The principle of P-Tuning v2 is to obtain a smaller and more efficient lightweight model by pruning the parameters of a trained large language model. Specifically, P-Tuning v2 first uses an adaptive pruning strategy to trim the parameters in a large language model to remove unnecessary redundant parameters in it. Then, for the pruned parameters, P-Tuning v2 uses a special compression method, which can compress the parameter size more efficiently and significantly reduce the number of total parameters for model fine-tuning. The schematic diagram of P-Tuning v2 is shown in Fig. 3.

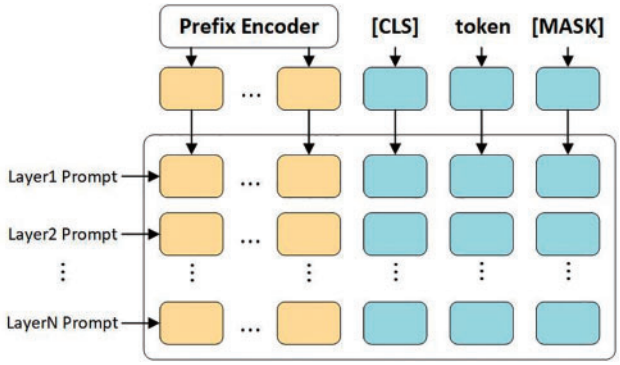

Figure 3: P-Tuning v2 schematic diagram

3.2.3 ChatGLM Model Training Details

Overall, the core idea of P-Tuning v2 is to make ChatGLM models lighter and more efficient, while keeping the performance of the models as unaffected as possible. This not only speeds up the training and inference of the model, but also reduces the consumption of memory and computational resources during the model's use, making the model more applicable to a variety of real-world application scenarios. The details of ChatGLM model training parameters are shown in Table 1.

Table 1: Training parameters

Parameters	Explanation	Setup of this article
PRE_SEQ_LEN	Length of soft prompt	128
LR	Learning rate of training	2e-2
Max_steps	Maximum training steps	3000
Max_source_length	Maximum input text length	64

(Continued)

Table 1 (continued)

Parameters	Explanation	Setup of this article
Train_batch_size	Training batch	1
Eval_batch_size	Validation batch	1
Gradient_accumulat-ion_steps	Steps of gradient accumulation	16
Logging_steps	Number of steps to print the log once	10
Save_steps	Number of steps to save the model once	1000
Quantization_bit	Model quantification	int8

3.3 Mining of Access Control Attribute Statements

This paper transforms the attribute mining issue into a sequence labeling problem involving subject attributes, object attributes, and action attributes. Primarily, the ID-CNN-CRF model is utilized for the sequence labeling task.

3.3.1 Access Control Statement Attribute Mining Model: ID-CNN-CRF

The Word embedding layer uses the BERT [31] model to transform the words in an access control statement into word vectors T = [T1, T2, ..., Tn]. The BERT model transfers a large number of operations traditionally done in downstream specific NLP tasks to the pre-trained language model, which further increases the generalization ability of the word vector model and adequately characterizes character-level, word-level, and sentence-level relationships. The BERT model is based on the bi-directional transformer technique for the training of the word vector model, which provides a deeper layer and better parallelism, and has very excellent performance in BERT model is trained based on bi-directional transformer technique for word vector modeling, which has deeper layers and better parallelism, and has excellent performance in many NLP tasks. Following the embedding process, the statements proceed to the IDCNN layer. This layer consists of one standard convolution and a dilated convolution module, with the latter comprising three layers of dilated convolutions with dilation rates of 1, 1, and 2. The standard convolutional neural network initially processes the features to form the input for the dilated convolution. Subsequent dilated convolutions are then performed, with the output of this module fed back into it repeatedly, creating a loop. Finally, the sequence is passed through a fully connected layer to yield the input for the CRF layer.

The CRF layer then learns the dependency relations between attribute labels in different words and outputs the most probable prediction tag sequence that adheres to the constraints of label transition.

The attribute mining model for access control statements is shown in Fig. 4.

3.3.2 DCNN

Traditional CNNs possess significant computational advantages; however, after convolution processes, peripheral neurons can capture only a fraction of the input text's information. To acquire contextual information, it necessitates the addition of more convolutional layers, resulting in a deeper network with an increasing number of parameters, which makes it prone to overfitting. Normally, CNN filters operate on a continuous segment of the input matrix, continually sliding to perform convolution, followed by pooling to integrate contextual information across multiple scales. However, this method can lead to a loss in resolution. Given that incorporating pooling layers results in

information loss and reduced accuracy, omitting them would decrease the receptive field, hindering the ability to learn global features. A straightforward solution of removing pooling layers and enlarging the convolution kernels increases computational demands significantly. At this juncture, the best solution is to employ dilated convolutions. The differences between traditional convolutions and dilated convolutions are depicted in Figs. 5 and 6.

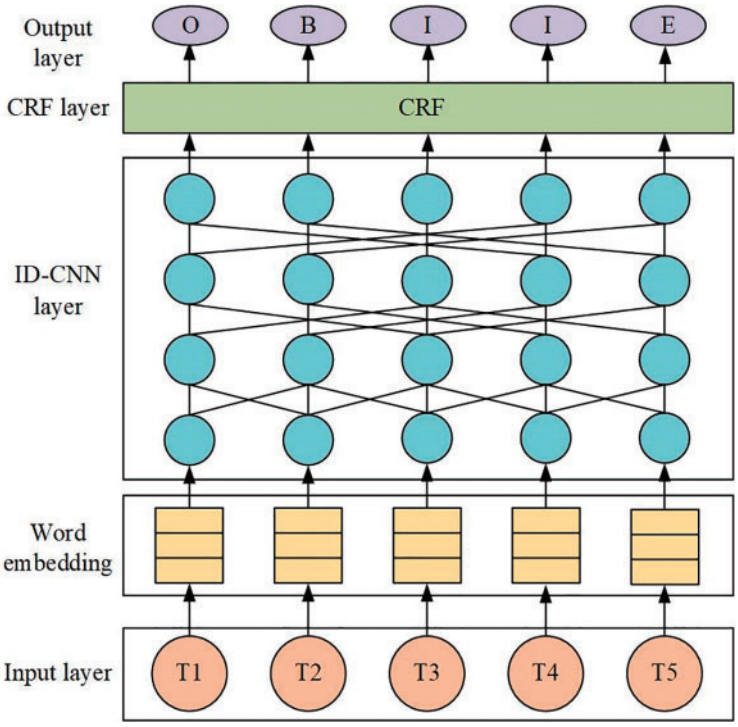

Figure 4: Access control statement attributes mining model

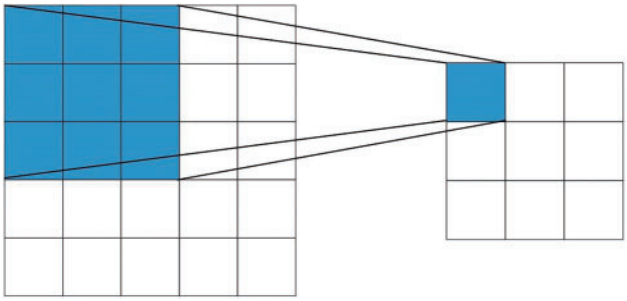

Figure 5: Traditional convolution

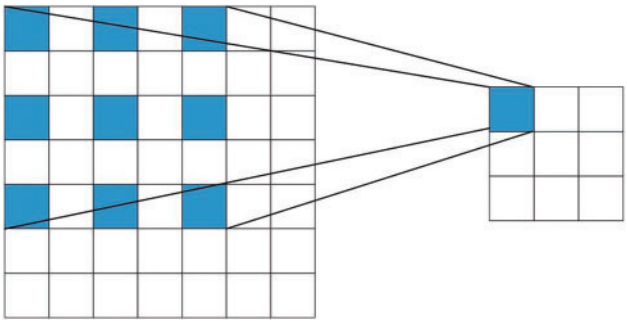

Figure 6: Dilated convolution

In dilated convolution, there are intervals between the elements of the convolution kernel, allowing it to cover a longer sequence of the input without increasing the size of the kernel or the number of parameters. As can be observed from Figs. 7 and 8, with a convolution kernel of size 3 and two layers of convolution, the dilated convolution has a context size of 7, whereas the traditional convolution has a context size of 5.

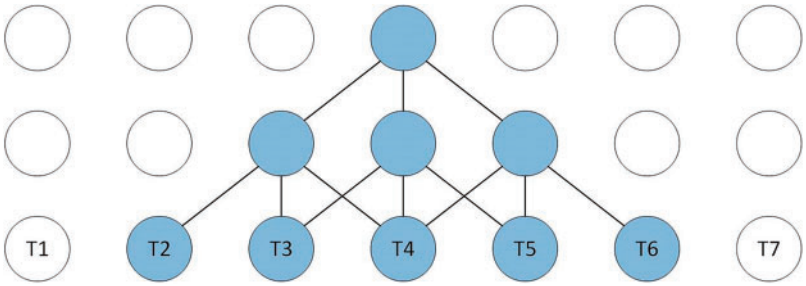

Figure 7: Traditional convolution over shorter input sequences

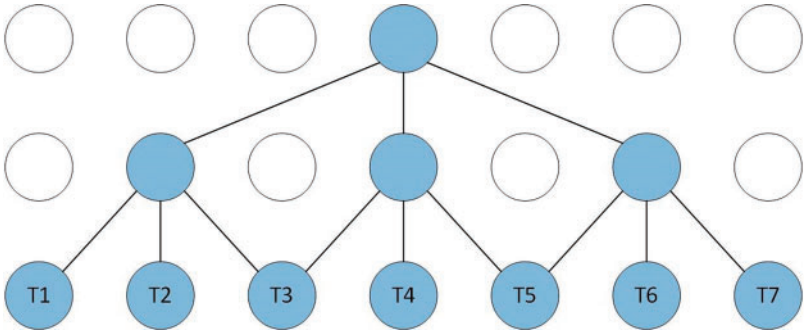

Figure 8: Dilated convolution spanning longer input sequences

3.3.3 ID-CNN

Stacking dilated convolution layers directly enables the capture of long-range contextual information; for instance, with 9 layers of dilated convolutions, the context width can exceed 1000. However, simply stacking multiple dilated convolution layers can easily lead to overfitting. To circumvent this, one can construct a dilated module with a modest number of layers, which contains several

dilated convolution layers. Data is then recurrently fed into the same module, meaning the output of the module is repeatedly input back into it. This model is known as an ID-CNN. By reusing the same module, the model can assimilate broader contextual information while maintaining good generalization capabilities.

The model initially takes in a vector sequence of length L, denoted as c_t, as its input and ultimately yields a sequence of the same length, noted as l_t. The i-th layer of dilated convolution with a dilation factor of ϕ is denoted as $K_{\phi}^{(i)}$. The first layer of the model is a regular convolution layer with $\phi = 1$, which transforms the input data into a new representation p_t. As shown in Eq. (1).

$$p_t = K_1^{(0)} c_t \tag{1}$$

Subsequently, a dilated module is constructed using n dilated convolution layers to process the input p_t, integrating increasingly broader contextual information into the embedded representation of each c_t. Represented by $R()$, the ReLU activation function serves in the transformation process. Starting from $m_t^{(0)} = p_t$, the stacked layers are recursively defined as shown in Eq. (2).

$$m_t^{(i)} = R\left(K_{2^n-1}^{(i-1)} m_t^{(i-1)}\right) \tag{2}$$

The final layer is represented as shown in Eq. (3).

$$m_t^{(n+1)} = R\left(K_1^{(n)} m_t^{(n)}\right) \tag{3}$$

The dilated module is denoted by $Q()$. In order to integrate broader contextual information without overfitting and without increasing the depth of Q, Q is applied iteratively s times. Starting with $q_t^{(1)} = Q\left(\mathrm{p}_t\right)$, the data is processed as shown in Eq. (4).

$$q_t^{(a)} = Q\left(q_t^{(a-1)}\right) \tag{4}$$

By subjecting this final representation to a simple affine transformation W_o, scores for each moment t of the sequence c_t belonging to different categories can be obtained in Eq. (5).

$$l_t^{(s)} = W_o q_t^{(s)} \tag{5}$$

3.3.4 OBIE Annotation Method

This article presents a design for an OBIE tagging method. The 'O' tag identifies attributes that are unrelated to access control statements. 'B' marks the beginning of an attribute, 'I' indicates the middle, and 'E' signifies the end of an attribute. The detailed tagging of subject attributes, object attributes, and action attributes is demonstrated in Table 2.

Table 2: Attribute tagging

Serial number	Tagging symbols	Tag types
1	/O	Irrelevant attributes
2	/B_subj	Subject attribute starting position
3	/I_subj	Subject attribute middle part
4	/E_subj	Subject attribute end position
5	/B_act	Action attribute starting position
6	/I_act	Action attribute middle part

(Continued)

Table 2 (continued)

Serial number	Tagging symbols	Tag types
7	/E_act	Action attribute end position
8	/B_obj	Object attribute starting position
9	/I_obj	Object attribute middle part
10	/E_obj	Object attribute end position

For the access control statement "Teachers and students can access their course information" using the OBIE tagging method, it would be tagged as:

/B_sub: Teachers /I_sub: and /E_sub: students /O: can /B_act: access /B_obj: their /I_obj: course /E_obj: information.

4 Experimental Evaluation

4.1 Dataset and Experimental Environment

The model presented in this article has been tested on a dataset listed in Table 3. There are four types of datasets: iTrust, Cyberchair, IBM [32] and Collected. iTrust is a patient-centric application dedicated to the maintenance of electronic health records. Cyberchair is a conference management system. IBM is a course management system for universities and colleges. Collected is an assortment of real-world policy documents that pertain to the authors' field, outlining security access permissions across various departments. This dataset is instrumental in evaluating the versatility and effectiveness of the model in diverse data management contexts. We combined the datasets from the four different domains into a total of 2283 sentences, of which 1724 were ACP sentences and 1114 were not ACP sentences.

Table 3: Dataset description

Document	Area	Number of ACP sentences	Number of Non-ACP sentences	Total
iTrust	Healthcare	902	619	1521
Cyberchair	Conference	219	302	521
IBM	Education	337	156	493
Collected	Multiple	266	37	303
Total	–	1724	1114	2838

The hardware and software environment for the experiments is as follows: Windows 11 64-bit, powered by a 12th Generation Intel(R) Core(TM) i7-12700H processor at 2.70 GHz, equipped with a GeForce GTX 3060 GPU, with 16 GB of memory. The software stack includes PyTorch version 1.12.1 and Python version 3.8.

4.2 Evaluation Criteria

To evaluate the results, we employ Precision, Recall, and F1 score metrics. Precision is the ratio of accurately identified ACP sentences, while Recall is the ratio of ACP sentences that have been

successfully retrieved. To compute these values, the classifier's predictions are categorized into four groups. True Positives (TP) are ACP sentences correctly predicted. True Negatives (TN) are non-ACP sentences correctly identified as such. False Positives (FP) are non-ACP sentences incorrectly labeled as ACP sentences. Lastly, False Negatives (FN) are ACP sentences that were not correctly identified as such. Utilizing these categorizations, the corresponding formulas for calculating Precision, Recall, and F1 score are as shown in Eqs. (6)–(8).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

4.3 Experimental Setup

The hyperparameters of the experiment are set as follows. The input layer uses the BERT model to transform the text into word vectors, the dimension of word embedding is 200, the number of convolution kernels is set to 240 in the IDCNN model, the dilation width of the three-layer dilation convolution of the expansion convolution module is 1, 1, 2, and the number of module loops is 4. The activation function used in the fully connected layer is the ReLU function. The batch_size is set to 8 and the epoch is 12 during the training process.

4.4 Baseline Models

To compare the performance of access control attribute mining across different neural network models, this study has selected three models as baseline comparators. The descriptions of the baseline models are as follows:

1. Bi-LSTM: A standalone Bi-LSTM neural network model designed for processing sequential data, such as text, speech, or time series data. The Bi-LSTM, an extension of the LSTM (Long Short-Term Memory) network, enhances the model's performance by integrating two LSTM layers—one processes the sequence information in a forward direction, and the other in a reverse direction. LSTMs introduce the concept of three gates (input gate, forget gate, and output gate) to regulate the flow of information, addressing the issues of vanishing or exploding gradients found in traditional RNNs. These gates control the preservation, updating, and forgetting of information, enabling LSTMs to maintain stable learning and memory capabilities over long sequences. Thus, the Bi-LSTM is capable of capturing both forward and backward dependencies within sequential data, significantly improving the model's understanding of sequences;
2. ID-CNN: An independent ID-CNN neural network model primarily utilized in the domain of Natural Language Processing (NLP). This model excels in tasks such as text categorization and entity recognition, adeptly handling long-range dependencies. ID-CNN employs dilated convolution to expand its receptive field and iteratively enhances its capabilities to capture long-term dependencies without the need to increase the network's depth or complexity;
3. Bi-LSTM-CRF: This model builds upon the Bi-LSTM neural network framework by incorporating a CRF (Conditional Random Field) layer. In labeling tasks, the label assigned to a word depends on the labels of neighboring words within the context. The CRF layer optimizes

globally, considering the joint probability distribution of the entire sequence of labels, which provides superior performance compared to models that optimize locally with independent label predictions.

4.5 Comparison and Analysis of Experimental Results

4.5.1 Comparison of Different Neural Network Models

All neural network models utilized the Word2Vec model to convert text into word vectors for input. As shown in Table 4 and Fig. 9, the ID-CNN-CRF model proposed in this study achieved precision and recall rates above 0.94, with the highest F1 score of 0.961 and the lowest loss value of 0.117 in the access control attribute mining experiments. The overall results demonstrate that the ID-CNN-CRF model's F1 score is 4.6% higher than that of the Bi-LSTM model, 4.1% higher than the ID-CNN model, and 0.7% higher than the Bi-LSTM-CRF model.

Table 4: Performance comparison in access control attribute mining

Model	Dataset	Precision	Recall	F1
Bi-LSTM	iTrust, Cyberchair, IBM, Collected	0.911	0.919	0.915
ID-CNN	iTrust, Cyberchair, IBM, Collected	0.918	0.923	0.920
Bi-LSTM-CRF	iTrust, Cyberchair, IBM, Collected	0.956	0.949	0.952
ID-CNN-CRF (Our)	iTrust, Cyberchair, IBM, Collected	0.962	0.961	0.961

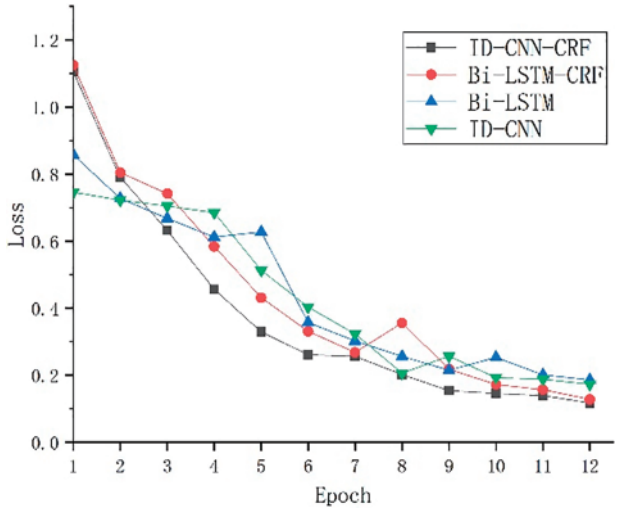

Figure 9: Changes in loss values across epochs for different models

The ID-CNN-CRF model presented in this paper shows superior local feature extraction ability from text when compared with the Bi-LSTM model due to its use of convolutional neural networks, along with a certain degree of fault tolerance to noisy data, thus resisting interference from such noise. Compared to the ID-CNN model, the addition of the CRF layer allows modeling of dependencies

The variations in F1 scores for both the ID-CNN-CRF model and the Bi-LSTM-CRF model across epochs are illustrated in Fig. 10. Although the differences in precision, recall, and F1 score between the ID-CNN-CRF model and the Bi-LSTM-CRF model in the context of mining access control attributes are minimal, the ID-CNN-CRF model outperforms the Bi-LSTM-CRF model in terms of speed by a factor of 1.48, operating faster under the same dataset and identical parameters, as shown in Table 5.

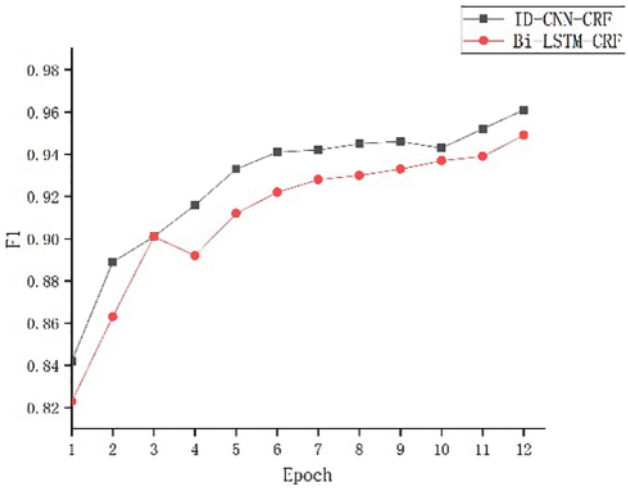

Figure 10: Variation of F1 scores across epochs for the two models

Table 5: Speed comparison under the same model parameters

Model	Embddding_dim	batch_size	epoch	speed
Bi-LSTM-CRF	200	8	12	1×
ID-CNN-CRF	200	8	12	1.48×

Due to the sequential nature of the Bi-LSTM-CRF model, the bidirectional processing of Bi-LSTM and the global optimization performed by the CRF layer result in high computational costs during training and inference. The complexity of the model prevents parallelization and does not leverage the full potential of GPU performance. In contrast, the ID-CNN-CRF model boasts a relatively simpler structure compared to the Bi-LSTM-CRF model. It is easier to implement and debug, and it can fully utilize GPU capabilities, thereby enhancing the model's execution speed.

4.5.2 Comparison with Other Literature Methods

A comparison of this paper's method with existing access control statement identification methods in terms of precision, recall and F1 score is shown in Table 6. The combined effect of the proposed model is the best compared with the existing methods, and the F1 value can reach 0.961, which is 4.6%

higher than the current state-of-the-art method. This is because the access control statements in natural language documents are first filtered using the ChatGLM model, which helps the subsequent labeling of access control attributes by the ID-CNN-CRF model. Moreover, the ID-CNN-CRF model expands the sensory field by inflating the convolution, which allows the model to receive wider contextual information, and enhances the model's ability to capture long-term dependencies by iterating this structure without increasing the depth or complexity of the network.

Table 6: Comparison results with other literature methods

Method	Dataset	Precision	Recall	F1
Document [19]	iTrust, Cyberchair, IBM, Collected	0.863	0.874	0.869
Document [23]	iTrust, Cyberchair, IBM, Collected	0.873	0.842	0.857
Document [24]	iTrust, Cyberchair, IBM, Collected	0.919	0.849	0.883
Document [25]	iTrust, Cyberchair, IBM, Collected	0.923	0.907	0.915
Our method	iTrust, Cyberchair, IBM, Collected	0.962	0.961	0.961

4.5.3 Ablation

In order to analyze and compare the contribution of each component in the model and further understand the role of each component in access control policy generation, we experimented with the entire access control policy generation framework model ChatGLM+ID-CNN-CRF with two changes on the dataset:

1. ChatGLM: Using the ChatGLM model alone, the access control attribute mining step is skipped and the access control attributes in natural language documents are extracted directly using the ChatGLM model;
2. ID-CNN-CRF: Using the ID-CNN-CRF model alone, the access control statement identification step is skipped and the access control attributes in natural language documents are labeled directly using the ID-CNN-CRF model.

The results of the comparison between the two modifications of the access control policy generation framework and the approach of this paper are shown in Table 7.

Table 7: Comparison of result of ablation experiments

Method	Dataset	Precision	Recall	F1
ChatGLM	iTrust, Cyberchair, IBM, Collected	0.763	0.721	0.741

(Continued)

Table 7 (continued)

Method	Dataset	Precision	Recall	F1
ID-CNN-CRF	iTrust, Cyberchair, IBM, Collected	0.823	0.839	0.831
ChatGLM+ID-CNN-CRF	iTrust, Cyberchair, IBM, Collected	0.962	0.961	0.961

From the results in Table 7, we can draw the following conclusions: 1) Using the ChatGLM model alone, the performance of directly extracting access control attributes from natural language documents is poor, because there is no OBIE attribute labeling method in the ID-CNN-CRF model to accurately label subject attributes, object attributes, and action attributes. It highlights the importance of ID-CNN-CRF model for access control attribute labeling. 2) Skipping the access control statement identification step, without the screening of access control statements by ChatGLM model, the direct use of ID-CNN-CRF model for labeling attributes from complex natural language documents is not accurate. These observations suggest that the complementary and synergistic effects of various modules are crucial for the accuracy of access control policy generation. By utilizing a combination of access control statement identification and access control attribute mining, excellent performance can be achieved on the dataset

4.5.4 The Impact of Different Annotation Methods on Performance

In addition to the OBIE annotation method used in Section 3.3.4, this paper also uses the OB annotation method for comparative experiments. In the OB annotation method, "O" is used to label irrelevant attributes, while "B" is used to label relevant attributes. The detailed labeling of subject attributes, object attributes, and action attributes is shown in Table 8.

Table 8: OB annotation method

Serial number	Tagging symbols	Tag types
1	/O	Irrelevant attributes
2	/B_sub	Subject attribute
3	/B_act	Action attribute
4	/B_obj	Object attribute

The impact of using different annotation methods on mining access control attributes is shown in Table 9. Under the conditions of using the same dataset and the same model ID-CNN-CRF, the OBIE annotation method adopted in this paper demonstrates superior performance across all metrics.

Table 9: Comparison of different annotation methods

Annotation method	Precision	Recall	F1
OB	0.892	0.906	0.899
OBIE	0.962	0.961	0.961

4.6 Discussion

In our research, during the extraction of access control statements, we conducted an experiment by changing the ChatGLM model parameter quantization_bit mentioned in Section 3.2.3 to int4. Compared to the int8 used in this paper, the int4 model is more lightweight and reduces the cost of deploying the model. However, the effectiveness of extracting access control statements was not as accurate as with int8. Therefore, we ultimately used the int8 parameter setting. Furthermore, the method proposed in this paper has potential scalability for handling large datasets. The ChatGLM model possesses robust computational power, and by adjusting parameters such as the length of the soft prompt, it can be sufficiently trained to handle more data and extract access control statements. The ID-CNN-CRF model can increase the dilation rate of dilated convolutions and the number of iterations, enabling better integration of context in large datasets, adding attention mechanisms to select key information, and annotating access control attributes. We plan to validate our ideas in our future work.

Despite the promising results shown by the proposed method, there are still some challenges to be addressed. The method needs further improvement in the security and quality of policy generation. We hope to draw inspiration from current emerging machine learning and artificial intelligence algorithms. In our future work, we plan to add a new module for security and quality detection of access control policies after their generation, to enhance organizational privacy security and the quality of access control policies.

5 Conclusions

This paper proposes a method based on deep learning to automatically generate attribute-based access control (ABAC) policies from natural language documents. We decompose the problem of generating access control policies into two tasks: extracting access control statements and mining attributes from those statements. We utilize the ChatGLM pre-trained model to extract access control statements from natural language documents. Subsequently, the ID-CNN-CRF model is employed to annotate subject attributes, object attributes, and action attributes in the extracted access control statements, thereby efficiently generating effective access control policies. The experimental results validate the efficacy of the proposed method, and we also explore the impact of different attribute annotation methodologies on model performance. The next step in the work will try to improve the security of access control policy generation and the quality of the generated access control policies, as well as to achieve faster and more efficient access control decisions in large systems and high concurrency scenarios.

Acknowledgement: The authors wish to express their appreciation to the reviewers for their helpful suggestions which greatly improved the presentation of this paper.

Funding Statement: This paper was supported by the National Natural Science Foundation of China Project (No. 62302540), please visit their website at https://www.nsfc.gov.cn/ (accessed on 18 June 2024); The Open Foundation of Henan Key Laboratory of Cyberspace Situation Awareness (No. HNTS2022020), Further details can be found at http://xt.hnkjt.gov.cn/data/pingtai/ (accessed on 18 June 2024); Natural Science Foundation of Henan Province Youth Science Fund Project (No. 232300420422), you can visit https://kjt.henan.gov.cn/2022/09-02/2599082.html (accessed on 18 June 2024).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Fangfang Shan, Zhenyu Wang; data collection: Zhenyu Wang, Mengyao Liu, Menghan Zhang; analysis and interpretation of results: Fangfang Shan, Zhenyu Wang; draft manuscript preparation: Zhenyu Wang; manuscript guidance and revision: Fangfang Shan. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The iTrust data used to support the findings of this study have been deposited on the website: http://agile.csc.ncsu.edu/iTrust/wiki/ (accessed on 1 April 2024). The Cyberchair data used to support this study have been deposited on the website: http://cyberchair.cs.utwente.nl (accessed on 6 April 2024).

Ethics Approval: Not applicable.

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

References

[1] V. C. Hu, D. Ferraiolo, and D. R. Kuhn, *Assessment of Access Control Systems*. Washington, DC, USA: US Department of Commerce, National Institute of Standards and Technology, Sep. 2006.

[2] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, and J. Voas, "Attribute-based access control," *Computer*, vol. 48, no. 2, pp. 85–88, 2015. doi: 10.1109/MC.2015.33.

[3] D. Servos and S. Osborn, "Current research and open problems in attribute-based access control," *ACM Comput. Surv.*, vol. 49, no. 4, pp. 1–45, 2017.

[4] D. Ferraiolo, J. Cugini, and D. R. Kuhn, "Role-based access control (RBAC): Features and motivations," in *Proc. 11th Annu. Comput. Secur. Appl. Conf.*, New Orleans, LA, USA, 1995, pp. 241–248.

[5] K. Vijayalakshmi and V. Jayalakshmi, "A study on current research and challenges in attribute-based access control model, intelligent data communication technologies and internet of things," *Lect. Notes Data Eng. Commun. Technol.*, vol. 101, pp. 17–31, 2022. doi: 10.1007/978-981-16-7610-9.

[6] P. Samarati and S. C. De Vimercati, "Access control: Policies, models, and mechanisms," in *Foundations of Security Analysis and Design*. Berlin, Heidelberg: Springer, 2000, vol. 2171, pp. 137–196.

[7] S. Osborn, R. Sandhu, Q. Munawer, and S. Security, "Configuring role-based access control to enforce mandatory and discretionary access control policies," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 2, pp. 85–106, 2000. doi: 10.1145/354876.354878.

[8] V. C. Hu *et al.*, "Guide to attribute based access control (ABAC) definition and considerations (draft)," Special Publication (NIST SP)–800–162, 2013.

[9] C. Cotrini, T. Weghorn, and D. Basin, "Mining ABAC rules from sparse logs," in *2018 IEEE Eur. Symp. Secur. Priv. (EuroS&P)*, London, UK, 2018, pp. 31–46.

[10] S. Das, S. Sural, J. Vaidya, and V. Atluri, "Using gini impurity to mine attribute-based access control policies with environment attributes," in *Proc. 23nd ACM Symp. Access Control Models Technol. (SACMAT'18)*, New York, NY, USA, Association for Computing Machinery, 2018, pp. 213–215.

[11] P. Iyer and A. Masoumzadeh, "Mining positive and negative attribute-based access control policy rules," in *Proc. 23nd ACM Symp. Access Control Models Technol. (SACMAT'18)*, New York, NY, USA, Association for Computing Machinery, 2018, pp. 161–172.
[12] S. Shang, X. Wang, and A. Liu, "ABAC policy mining method based on hierarchical clustering and relationship extraction," *Comput. Secur.*, vol. 139, 2024, Art. no. 103717. doi: 10.1016/j.cose.2024.103717.
[13] A. Bamberger and M. Fernández, "Automated generation and update of structured ABAC Policies," in *Proc. 2024 ACM Workshop Secur. Trustworthy Cyber-Phys. Syst. (SaT-CPS'24)*, New York, NY, USA, Association for Computing Machinery, 2024, pp. 31–40.
[14] M. W. Sanders and C. Yue, "Mining least privilege attribute based access control policies," in *Proc. 35th Annu. Comput. Secur. Appl. Conf. (ACSAC'19)*, New York, NY, USA, Association for Computing Machinery, 2019, pp. 404–416.
[15] D. Mocanu, F. Turkmen, and A. Liotta, "Towards ABAC policy mining from logs with deep learning," in *18th Int. Multiconf., IS2015, Intell. Syst.*, Ljubljana, Slovenia, 2015.
[16] Z. Xu and S. D. Stoller, "Mining attribute-based access control policies from logs," in *Data Appl. Secur. Priv. XXVIII: 28th Annu. IFIP WG 11.3 Work. Conf., DBSec 2014*, Vienna, Austria, Springer, Jul. 14–16, 2014, pp. 276–291.
[17] L. Karimi, M. Aldairi, J. Joshi, and M. Abdelhakim, "An automatic attribute-based access control policy extraction from access logs," *IEEE Trans. Dependable Secur. Comput.*, vol. 19, no. 4, pp. 2304–2317, Jul. 1–Aug. 2022. doi: 10.1109/TDSC.2021.3054331.
[18] W. Ammar, S. Wilson, N. Sadeh, and N. A. Smith, "Automatic categorization of privacy policies: A pilot study," 2012. Accessed: May 11, 2024. [Online]. Available: http://reports-archive.adm.cs.cmu.edu/anon/isr2012/CMU-ISR-12-114.pdf
[19] X. Xiao, A. Paradkar, and T. Xie, "Automated extraction and validation of security policies from natural-language documents," in *Proc. ACM SIGSOFT 20th Int. Symp. Found. Softw. Eng. (FSE '12)*, New York, NY, USA, Association for Computing Machinery, 2012, pp. 1–11.
[20] Z. Zhu, Z. Ren, and X. Du, "Unstructured text ABAC attribute mining technology based on deep learning," in *2021 3rd Int. Academic Exch. Conf. Sci. Technol. Innov. (IAECST)*, Guangzhou, China, 2021, pp. 34–39.
[21] Y. Xia, S. Zhai, Q. Wang, H. Hou, Z. Wu and Q. Shen, "Automated extraction of ABAC policies from natural-language documents in healthcare systems," in *2022 IEEE Int. Conf. Bioinform. Biomed. (BIBM)*, Las Vegas, NV, USA, 2022, pp. 1289–1296.
[22] J. Slankas and L. Williams, "Access control policy extraction from unconstrained natural language text," in *2013 Int. Conf. Soc. Comput.*, Alexandria, VA, USA, 2013, pp. 435–440.
[23] M. Narouei, H. Khanpour, H. Takabi, N. Parde, and R. Nielsen, "Towards a top-down policy engineering framework for attribute-based access control," in *Proc. 22nd ACM Symp. Access Control Models Technol. (SACMAT'17 Abstr.)*, New York, NY, USA, Association for Computing Machinery, 2017, pp. 103–114.
[24] M. Narouei, H. Takabi, and R. Nielsen, "Automatic extraction of access control policies from natural language documents," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 3, pp. 506–517, May 1–Jun. 2020.
[25] M. Alohaly, H. Takabi, and E. Blanco, "A deep learning approach for extracting attributes of ABAC policies," in *Proc. 23nd ACM Symp. Access Control Models Technol. (SACMAT'18)*, New York, NY, USA, Association for Computing Machinery, 2018, pp. 137–148.
[26] S. Zhao, L. A. Tuan, J. Fu, J. Wen, and W. Luo, "Exploring clean label backdoor attacks and defense in language models," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 32, pp. 3014–3024, 2024.
[27] W. You, Z. Hammoudeh, and D. Lowd, "Large language models are better adversaries: Exploring generative clean-label backdoor attacks against text classifiers," in *Findings Assoc. Comput. Linguist.: EMNLP 2023*, Singapore, Association for Computational Linguistics, 2023, pp. 12499–12527.
[28] M. Shao *et al.*, "An empirical evaluation of llms for solving offensive security challenges," 2024, *arXiv:2402.11814*.

[29] Z. Du *et al.*, "GLM: General language model pretraining with autoregressive blank infilling," in *Proc. 60th Annu. Meet. Assoc. Comput. Linguist.*, Dublin, Ireland, Association for Computational Linguistics, 2022, pp. 320–335.

[30] X. Liu *et al.*, "P-Tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks," in *Proc. 60th Annu. Meet. Assoc. Comput. Linguist.*, Dublin, Ireland, Association for Computational Linguistics, 2022, pp. 61–68.

[31] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North Am. Chapter Assoc. Comput. Linguist.: Hum. Lang. Technol.*, Minneapolis, Minnesota, Association for Computational Linguistics, 2019, pp. 4171–4186.

[32] J. Slankas, X. Xiao, L. Williams, and T. Xie, "Relation extraction for inferring access control rules from natural language artifacts," in *Proc. 30th Annu. Comput. Secur. Appl. Conf. (ACSAC'14)*, New York, NY, USA, Association for Computing Machinery, 2014, pp. 366–375.