Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.049011

ARTICLE

Big Data Access Control Mechanism Based on Two-Layer Permission Decision Structure

Aodi Liu, Na Wang*, Xuehui Du, Dibin Shan, Xiangyu Wu and Wenjuan Wang

He'nan Province Key Laboratory of Information Security, Information Engineering University, Zhengzhou, 450000, China

*Corresponding Author: Na Wang. Email: twftina_w@126.com

Received: 25 December 2023 Accepted: 08 March 2024 Published: 25 April 2024

ABSTRACT

Big data resources are characterized by large scale, wide sources, and strong dynamics. Existing access control mechanisms based on manual policy formulation by security experts suffer from drawbacks such as low policy management efficiency and difficulty in accurately describing the access control policy. To overcome these problems, this paper proposes a big data access control mechanism based on a two-layer permission decision structure. This mechanism extends the attribute-based access control (ABAC) model. Business attributes are introduced in the ABAC model as business constraints between entities. The proposed mechanism implements a two-layer permission decision structure composed of the inherent attributes of access control entities and the business attributes, which constitute the general permission decision algorithm based on logical calculation and the business permission decision algorithm based on a bi-directional long short-term memory (BiLSTM) neural network, respectively. The general permission decision algorithm is used to implement accurate policy decisions, while the business permission decision algorithm implements fuzzy decisions based on the business constraints. The BiLSTM neural network is used to calculate the similarity of the business attributes to realize intelligent, adaptive, and efficient access control permission decisions. Through the two-layer permission decision structure, the complex and diverse big data access control management requirements can be satisfied by considering the security and availability of resources. Experimental results show that the proposed mechanism is effective and reliable. In summary, it can efficiently support the secure sharing of big data resources.

KEYWORDS

Big data; access control; data security; BiLSTM

1 Introduction

The continuous development of cloud computing, the Internet of Things, and other emerging technologies in recent years has contributed toward the generation of massive data resources. Thus, the onset of the era of big data [1,2] has led to significant changes in all aspects of industry and society. Big data is widely used in medicine, energy, finance, education, and other fields [3–5]. Data has become an important asset that can flow. The analysis and utilization of big data resources can create social and economic value; the larger the data volume and the wider the source, the larger the value. However, although big data provides new development opportunities, it also faces severe security challenges. For example, in March 2018, it was revealed that Facebook suffered a serious data breach. Specifically, Cambridge Analytica illegally accessed the personal data of more than 50 Facebook million users without their authorization and used the data to build mathematical models for analyzing citizens' political preferences. In May 2018, Under Armour revealed that an attacker had illegally accessed the information of more than 150 million users of MyFitnessPal—a platform that is used to track diet and exercise. In February 2019, India's state-owned gas company revealed that a database containing the biometric information of millions of users had been illegally accessed owing to a security breach. Thus, unauthorized sharing of big data poses a security threat to users. Realizing safe and controllable big data resource circulation and sharing is a fundamental requirement for big data application and development.

As an important means to protect data security, access control technology [6,7] manages users' permissions. It enables legitimate users to access the corresponding resources in the system according to their permissions and prevents unauthorized access by illegal users. Thus, it can effectively guarantee data security and normal operation of business systems. Therefore, effective access control measures are urgently required to ensure the safe and controllable sharing of big data resources. However, in the era of big data, access control faces some new challenges [8]. Big data resources are characterized [9] by large scale, wide sources, and strong dynamics, which makes the management scenarios more complex and the security requirements more diverse. The current access control methods can be classified according to the different bases of access control permission decisions. Discretionary access control and mandatory access control are early access control mechanisms. Discretionary access control directly establishes the user-permission relationship between users and resources. It is flexible, but the permission control is scattered and difficult to manage. Mandatory access control is implemented according to the subject-object security tag level. It has strong permission control ability but lacks flexibility and does not support fine-grained access control. Role-based access control implements control based on the operation permissions of subject roles and objects. It simplifies authorization operations and is easier to maintain, but it is difficult to support large-scale dynamic and fine-grained access control. Hence, the above mechanism is mainly applied to the closed computing environment with limited resource scale, which is difficult to apply to big data environments. With the ever-changing demand for access control, in recent years, researchers have introduced risk [10], intent [11], space-time [12,13], behavior [14] and other factors to achieve access control in different scenarios. With the increasing demand for large-scale dynamic and fine-grained access control, attribute-based access control (ABAC) [15] is proposed. The ABAC uses attributes as the basic elements of access control, which can realize a more accurate description of the subject and object, and has the advantages of strong semantic expression ability, high flexibility, and strong expansibility. Access control elements in other access control mechanisms, such as security tags, roles, risks, intents, and space-time, can be abstracted into specific types of attributes in ABAC.

Existing research on attribute-based access control uses attributes to describe entities, and uses policies to describe attribute-permission correspondence [16]. In this way, access control policies and attributes together provide support for the implementation of access control. Security managers configure policies based on their professional knowledge, such as security requirements and service requirements. These policies consist of static rules and do not have the ability to evolve dynamically. When the access control service receives the access request, it determines the permission strictly according to the entity attribute and policy. In a big data scenario, the scale of attributes and policies increases with the increase of users and resources. It is increasingly difficult for security managers to implement effective and comprehensive policy management manually. Therefore, it is necessary to endue the traditional static strategy with dynamic and automatic adaptability. Due to the complexity of big data applications and the diversity of security requirements, big data access control presents a new trend and characteristics of diversified decision basis, fuzzy decision results and integration of multiple access control technologies. New requirements for automatic and adaptive access control have been proposed. The present study focuses on the following two challenges of big data access control.

(1) It is difficult for security managers to conduct efficient policy management of massive and dynamic big data resources based on professional knowledge, which poses a challenge to the implementation of dynamic access control. In existing access control methods, the management of policies is inseparable from specific application scenarios. It is necessary for security managers to manually formulate access control policies on the basis of their professional knowledge to protect data resources. In a closed environment, it is safe and feasible to manage a policy manually with limited data resources. However, in the open big data environment, it is tedious to manually manage massive and dynamic data resources. Therefore, an access control mechanism for big data resources needs to be automatic and adaptive to improve the efficiency of permission management.

(2) It is difficult to accurately describe the access control policy of big data resources gathered from multiple sources, which poses a challenge to the implementation of fine-grained access control. The value density of big data resources is low. The core value does not lie in a single data resource itself, and analysis results show a significantly higher value density after the analysis of massive data. Furthermore, the characteristics of multi-source convergence and cooperative sharing of big data make policy management more difficult, leading to the increasingly serious phenomena of excessive authorization and insufficient authorization. How to determine "which users are allowed to access which resources" is an important professional problem that is difficult to accurately describe in the context of big data. Systems often adopt excessive authorization to maintain their availability. In addition, owing to application complexity, some new access requirements are often not considered in advance by security managers. The phenomenon of insufficient user authorization is becoming increasingly common from the viewpoint of protecting resources. Thus, it is difficult to achieve a trade-off between the security and availability of resources.

The existing extended research on the ABAC model includes introducing extended attribute elements such as behavior, negative attribute, task, trust and risk into the model to improve the expression ability of the ABAC model. Different extended attributes have different capabilities and advantages in different scenarios. However, the ability to describe the content of data resources is still lacking. To address the above-mentioned challenges of big data access control, this research extends the existing ABAC model and introduces the concept of business attributes in the ABAC model to constrain the business behavior of entities. A big data access control mechanism based on a two-layer permission decision structure is proposed. The inherent attributes and business attributes of access control entities are combined to form a two-level permission decision structure, which combines accurate decisions based on static policy with fuzzy decisions based on business constraints. This method can implement adaptive and intelligent policy management according to the similarity calculation between the business attributes of subjects and resources. The method considers both the security and availability of resources and provides effective support for the secure sharing of big data resources. The main contributions of our research are as follows:

(1) This research proposes a big data access control mechanism based on a two-layer permission decision structure (BDAC-TPDS). This model introduces the concept of business attributes and implements a two-layer hybrid access control structure composed of the inherent attributes and business attributes of entities.

(2) This research implements a general permission decision algorithm based on logical calculation for the inherent attributes of entities. Based on the logical relationship between the inherent attributes of subjects and resources, it can make accurate permission decisions.

(3) This research implements the business permission decision algorithm based on a bi-directional long short-term memory (BiLSTM) neural network, which calculates the similarity between the business attributes of subjects and resources to make permission decisions.

Through the two-layer mixed access control structure, we can flexibly implement permission decisions with different strengths and policy management with different granularities. The remainder of this paper is organized as follows. Section 2 reviews the related work. Session 3 formalizes the adaptive access control model based on business constraints and introduces the implementation framework in detail. Session 4 proposes a two-layer permission decision algorithm and describes the details of the algorithm. Session 5 analyzes the security and availability of the model. Session 6 describes experiments conducted to verify the effectiveness of the mechanism and discusses the experimental results. Finally, Section 7 concludes the paper.

2 Related Work

Although existing research on big data access control remains in its infancy, some progress has been made in this field. In order to improve the adaptive ability of access control of massive data resources, it mainly includes trust-based access control, risk-based access control, and semantic-based access control. Trust-based access control principle determines whether an authorization requester is allowed to access a resource based on its trust level. That is, only the requester with a certain trust level can obtain access authorization. Zhao et al. [17] proposed a zero trust access authorization and control of network boundary based on cloud big data fuzzy clustering. The fuzzy clustering algorithm of cloud big data is used to mine the data related to user behavior, obtain the user's trust level by designing a trust evaluation mechanism, and then implement dynamic access control by combining the user's trust level to distinguish legitimate requests from illegal requests, and complete the zero-trust access authorization and control at the network boundary. To solve the access control problem in the fog computing environment, Daoud et al. [18] proposed an access control service protocol based on trust evaluation and user activity monitoring. The trust level of the user is calculated and measured by the access history of the user, the type of host the type of user. Then, the certificate is authorized for each user that meets the requirements for the trust level. Users with authorization certificates will not repeat the access process when they access the node. Jiang et al. [19] proposed a trust-role-based access control (T-RBAC) model based on two-dimensional dynamic trust evaluation. The model uses analytic hierarchy process and grey theory to quantify the role attribute trust and uses Euler measurement and probability statistics to quantify the user behavior trust in the historical behavior dimension. Then the trust knowledge base performs hierarchical authorization according to the weighted average comprehensive trust value. Subsequently, Jiang et al. [20] proposed an access control model based on user credibility to solve the problem that existing methods could not describe the timeliness and trend of quantized values, which may not reflect the real trust situation well. This model also quantifies trust based on user history access records but introduces user history behavior trends into trust evaluation model through the corresponding regression analysis model. Chen et al. [21] divided user trust into two parts: Attribute trust and historical trust. Attribute trust is calculated according to the attributes of users, and historical trust is calculated according to the historical access behavior of users. Risk-based access control is a technique that uses risk assessment to decide whether to grant or deny access requests from users or systems. The principle is to assess the risk of a system, network or application according to security policies and security requirements and apply the risk assessment results to access control decisions. For the medical big data scenario, Jiang et al. [10] proposed an access control model based on spectral clustering (SC) and risk (SC-RBAC). The improved SC algorithm was adopted in this model to cluster doctors' users, and user classification was introduced into information entropy as a parameter, thus improving the accuracy of quantifying user access behavioral risks. Finally, access permissions are assigned to users based on the risk value of access behavior and the constructed access control function. Jiang et al. [22] introduced information entropy to quantify the access request risk and privacy risk when doctors access clinical data. Based on the assumption of bounded rationality, a multi-person evolutionary game model of risk access control is constructed. The dynamic selection strategy and evolutionary stability of participants can be analyzed, and the risk of doctor's visit behavior can be included in the profit function of evolutionary game, thus realizing the risk-adaptive access control. Ma et al. [23] proposed a risk-aware topic-based access control model (RTBAC), which uses topic to represent the content relationship between users and data, and uses risk techniques to grant users access permissions based on their historical behaviors and access requests. Semantic-based access control uses semantic information to represent the attributes of resources and users, and ontological knowledge to represent access policies and access control rules, so as to achieve more sophisticated, flexible and intelligent access control. Drozdowicz et al. [24] proposed an access control method that combines XACML (eXtensible Access Control Markup Language) policy with semantic reasoning. This approach allows "mix and match" (depending on the circumstances of the system being developed) XACML rules with properties derived from semantic reasoning, separating expert-made system rules from user preferences, and each user can dynamically make their own rules that represent individual attitudes toward privacy. Verginadis et al. [25] proposed a context-aware access control framework that combines effective context-aware access control policies and infuses them into cloud applications. It has a semantic reasoning function and supports minimal human intervention and work.

Due to the complexity of big data applications and the diversity of security requirements, the current access control mechanism makes it difficult to balance the security and availability of resources. Big data access control presents a new trend and characteristics of diversified decision basis, fuzzy decision results and integration of multiple access control technologies. Existing access control mechanisms are difficult to get rid of the dependence on static policies, which limits the dynamic and adaptive nature of access control. When the access control service receives the access request, it determines the permission strictly according to the attribute and policy. As the number of users and resources increases, it becomes more and more difficult for security managers to implement effective and comprehensive policy management manually. Therefore, it is necessary to endue the access control mechanism with dynamic and automatic adaptability and achieve accurate semantic-level access control. In order to solve this problem, it is necessary to improve the existing access control mechanism and combine precise decisions based on static policy with fuzzy decision based on business constraints to provide effective support for secure sharing of big data resources.

3 BDAD-TPDS Model

To impart the ABAC model with stronger dynamics and adaptive access control ability, this study extends the concept of entity attribute and introduces the concept of business attribute. A big data access control mechanism based on two-layer permission decision structure (BDAC-TPDS) is proposed, and a two-layer hybrid access control structure consisting of entity inherent attributes and business attributes is implemented.

3.1 Model Formulation

Definition 1 Entity attribute is used to describe the information of entity characteristics involved in the access control process. It is the core concept of the BDAC-TPDS model. Entity attributes are identified by attribute name and attribute value (Attribute: AttName=value). The process of assigning values to attributes is called attribute assignment. For convenience, we abbreviate entity attributes as attributes hereafter.

Definition 2 Inherent attribute is used to describe the inherent constraint information for entity access control. The quaternions (IS, IR, IE, IA) are used to represent the inherent attributes of subject, resource, environment, and action, which correspond to the concept of attributes in the classical ABAC model.

Definition 3 Business attribute is used to describe the business constraint information contained in subjects and resources. Binary groups (BS, BR) are used to represent subject business attributes and resource business attributes. Subject business attribute is used to describe the characteristics of relevant business resources that the subject can access. Resource business attribute is used to describe the business characteristics of relevant resources. Business attributes can be obtained through attribute mining technology [26,27].

Definition 4 Attribute tuple is a collection of attributes used to describe a particular entity. It is expressed as X (A1, A2, ..., An). An attribute is relatively static and stable, while an attribute tuple is dynamic and changes over time.

Definition 5 Attribute-based access request is a description of the visitor of the resource, the accessed resource, and the requested operation. It consists of a set of attribute tuples that contain at least one subject attribute, one resource attribute, and one action attribute. It can be expressed as AAR: {XS, XR, XA, XE}.

Definition 6 Access control policy is the constraint on the access ability of the subject and a concrete embodiment of the subject's authorization behavior toward the resource. It can be represented as ACP = {PID, IAS, IAR, action, sign}, where PID denotes the policy ID and sign $\in$ {permit, deny} indicates whether access is permitted or denied.

Definition 7 Policy decision is based on whether the entity attributes meet the constraint of the access control policy to make a decision as to whether the subject can access the corresponding resources. It can be expressed as PD = {IAS, BAS, IAR, BAR, action, Decision (IAS, IAR, ACR) && Verify (BAS, BAR)} Decision() and Verify() are used to determine whether inherent attributes and business attributes can meet the access conditions, respectively. If both Decision() and Verify() are satisfied, the subject can access the corresponding resource.

3.2 Model Description

Business constraints and user intent are described by business attributes in BDAC-TPDS. Business constraint is determined by the business attributes of the resource, while user intent is determined by the business attributes with which the subject is authorized. The access control framework is shown in Fig. 1. It includes the policy enforcement point (PEP), attribute authority (AA), policy administration point (PAP), and access decision point (ADP). AA includes the inherent attribute authority (AAI) and business attribute authority (AAB), while ADP is divided into the policy decision point (PDP) and business decision point (BDP). As shown in Fig. 1, the access control workflow of BDAC-TPDS can be divided into two phases: The preparation stage and the execution stage.

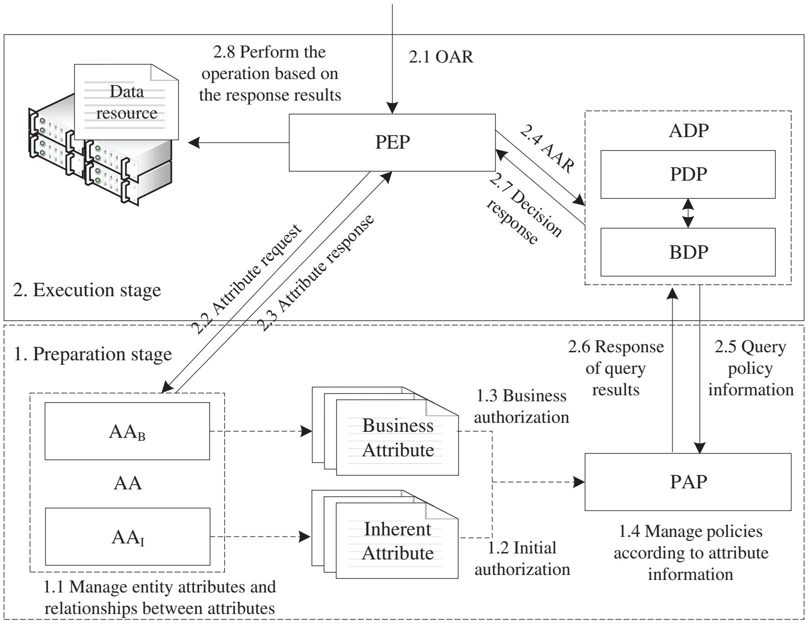

2.1 OAR
2.8 Perform the operation based on the response results
Data resource
PEP
2.4 AAR
2.7 Decision response
ADP
PDP
BDP
2.2 Attribute request
2.3 Attribute response
2. Execution stage
1. Preparation stage
2.6 Response of query results
2.5 Query policy information
AA_B
AA
AA_I
Business Attribute
Inherent Attribute
1.3 Business authorization
1.2 Initial authorization
PAP
1.4 Manage policies according to attribute information
1.1 Manage entity attributes and relationships between attributes

Figure 1: Access control framework of BDAC-TPDS

The preparation stage involves the management of entity attribute information and access control policy information involved in the access control process. It includes initial authorization and business authorization. a) In the initial authorization stage (also known as general authorization), the security manager sets the access control policy of the system according to the user's inherent attribute information. Users can access the resources on the basis of the initial authorization. However, the initial authorization is based only on the user's inherent attribute information (such as role, unit, position, gender), which is relatively long-term and static. The granularity of access control based on the inherent attribute information is too coarse. To achieve finer granularity and business-adaptive access control, business authorization will be performed dynamically after the initial authorization. b) In the business authorization stage, different types of business resources are characterized by different types of business attributes, and business authorization is realized by dynamically granting business attributes to users.

The execution stage deals with the decision, response, and execution of access requests. a) Original access request (OAR) is accepted by PEP as the original request for access to a specific resource sent by the user. PEP analyzes the semantics of subjects, resources, and action attributes in NAR. According to the attribute information obtained by AA, the processing access request AAR is generated and sent to ADP from PEP. b) ADP queries PAP for access control policy sets related to the requested big data resource. PDP makes a general decision of access control according to the policy set. If the decision result is denied, the result will be sent directly back to PEP. If the decision result is permitted, AAR will be sent to BDP. c) After receiving AAR, BDP calculates the similarity of the business constraints according to the authorized business attributes of the subject and the business attributes of the accessed resource. If the calculation result is similar, the decision result is permitted. If the calculation result is non-similar, the decision result is denied, and the result is sent back to PEP. d) PEP responds to the user's access request on the basis of the ADP decision result.

A two-layer hybrid access control structure constituted by two-layer authorization in the preparation stage and two-layer permission decision in the execution stage can realize access control of big data resources with finer granularity and greater relevance to business intentions.

3.3 Description of Inherent Attributes and Business Attributes

Inherent attribute is used to describe inherent constraint information of the access control entity. It includes multiple types of subject attributes (role, department, job, gender, age, credit, etc.), resource attributes (security level, risk, owner, creation time, etc.), and environment attributes. The use of inherent attributes facilitates a clear and precise description of entities and permissions. For example, the policies in a medical information system are summarized in Table 1.

Table 1: Examples of policies

Number	The content of policies
1	{(department = surgery, position = chief physician), (resource_type = case resource), (action = read), (sign = permit)}
2	{(doctorID = 0010), (resource_type = case resource, resourceID = 0003), (action = write, read), (sign = permit)}

According to policy 1, when the subject's department is surgery and the position is chief physician, the case resource can be read. According to policy 2, a doctor with a doctorID of 0010 can write and read the case resource with a resourceID of 0003. This permission management is extremely precise. However, the form of authorization in policy 1 is extremely coarse-grained, and the authorized user will be able to read all the case information. By contrast, the form of authorization in policy 2 is extremely fine-grained. It is difficult for security administrators to carry out such labor-intensive authorization with massive resources. In this case, the relatively fuzzy policy of "doctors can access case resources related to their research direction" is more consistent with the practical application requirements under the condition of big data. Access control entities and permissions can be vaguely described using business attributes. The simplest form is case 1 in Table 2. Physicians specializing in cardiovascular disease, cardiopathy, hypertension, and congenital heart disease will be able to access the case resources whose business attributes are cardiovascular disease, cardiopathy, hypertension, and congenital heart disease. In this case, the role of the business attributes is equivalent to that of inherent attributes. For case 2 in Table 2, physicians specializing in cardiovascular disease, cardiopathy, hypertension, and congenital heart disease can also access similar semantic case resources such as angina, cardiac failure, myocarditis, heart failure, arrhythmia, and sudden cardiac death. The business attributes of the subject and resource need not be completely consistent but only semantically similar (or comprise the vocabulary of the same domain) so that access permission can be granted. At the same time, the business attribute information of the resource does need not rely on strict and accurate manual management by the security manager. Automatic mining and extraction of resource business attributes can be realized by intelligent methods [26,27] that can effectively reduce the workload of attribute management and permission management.

Table 2: Examples of attributes

Number	Business attributes of subject	Business attributes of resource
1	(cardiovascular disease, cardiopathy, hypertension, congenital heart disease)	(cardiovascular disease, cardiopathy, hyper-tension, congenital heart disease)
2	(cardiovascular disease, cardiopathy, hypertension, congenital heart disease)	(angina, cardiac failure, myocarditis, heart failure, Arrhythmia, sudden cardiac death)

Thus, we can transform the permission decision problem of the subject and resource into the problem of semantic similarity matching between the business attribute sets of the subject and resource. As BiLSTM neural networks have been shown to perform well in the field of natural language processing, this study employs a BiLSTM network to realize permission decisions based on business attribute similarity (see Section 4.3 for details). Although permission decision based on business attributes can improve the efficiency of attribute and policy management, the decision accuracy is not as high as that in the case of inherent attributes. In particular, in scenarios where precise access control is required (such as data resources with high security levels), the role of precise permission management based on inherent attributes is irreplaceable. Therefore, we propose a two-layer permission decision structure that combines inherent attributes with business attributes. The structure can fully exploit the advantages of the two methods and overcome the shortcomings of each method. In the next section, we will elaborate on the process of two-layer permission decision.

4 Two-Level Permission Decision Algorithm

4.1 Permission Decision Process

The process of permission decision is shown in Fig. 2. First, a general authorization decision is made on the basis of logical calculation. If the decision result is denied, it is directly returned. If the decision result is permitted, the business permission decision algorithm based on the BiLSTM network will be executed. The business permission decision result is the final result.

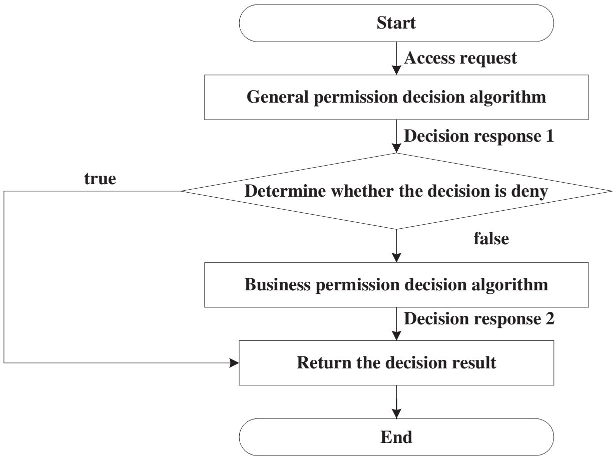

Start
Access request
General permission decision algorithm
Decision response 1
true
Determine whether the decision is deny
false
Business permission decision algorithm
Decision response 2
Return the decision result
End

Figure 2: Permission decision process

4.2 General Permission Decision Algorithm Based on Logical Calculation

In BDAC-TPDS, a general permission decision based on logical calculation is processed by the PDP module. The sign of the access control policy is a binary token of PERMIT or DENY. If the attribute information involved in the access request AR satisfies the logical constraint in the access control policy, the access request AR is responded to according to the policy sign (response result is PERMIT or DENY). If access request AR does not have a matching policy, it indicates that the policy and attribute information is insufficient to make a decision response. The response result will be UNKNOWN. Therefore, there are three results, namely PERMIT, DENY, and UNKNOWN, in BDAC-TPDS. We use ATTR_SETAR to denote the set of associated attributes involved in access request AR and ATTR_SETACP to denote the set of associated attributes involved in access control policy ACP. The general permission decision can be expressed as follows:

$$Decision\,(\mathrm{ATTR\_SET}_{AR}, \mathrm{ATTR\_SET}_{ACP}) \rightarrow \{\mathrm{PERMIT}, \mathrm{DENY}, \mathrm{UNKNOWN}\} \tag{1}$$

If all the inherent attributes in ATTR_SETAR conform to the constraint condition of the policy attribute in ATTR_SETACP, i.e., ATTR_SETAR ⊆ ATTR_SETACP, then:

$$Decision\,(ATTR\_SET_{AR}, ATTR\_SET_{ACP}) = ACP.sign \tag{2}$$

If the inherent attributes in ATTR_SETAR cannot all meet the constraints of the policy attribute in ATTR_SETACP, i.e., ATTR_SETAR ⊈ ATTR_SETACP, then:

$$Decision\,(ATTR\_SET_{AR}, ATTR\_SET_{ACP}) = \mathrm{UNKNOWN} \tag{3}$$

The algorithm is as follows:

Algorithm 1: Common permission decision algorithm

```
Input: Access control policy set, ACP_Set; Access request, AR
Output: Result of permission decision, Result(PERMIT, DENY, UNKNOWN)
  Begin algorithm
  PERMIT_RESULT, DENY_RESULT, UNKNOWN_RESULT = Null
  For i = 1 to ACP_Set. Length do
     result = Decision(AR, ACP_Set [i])
     if result = PERMINT then
           PERMIT_RESULT. add(ACP_Set [i].PID)
     else if result = DENY then
           DENY_RESULT. add(ACP_Set [i].PID)
     else then
           UNKNOWN_RESULT. add (ACP_Set [i].PID)
if (PERMIT_RESULT $\neq$ Null) && (DENY _RESULT==Null) then
           return PERMIT
else if (PERMIT_RESULT==Null) && (DENY _RESULT $\neq$ Null) then
     return DENY
else then
     return UNKNOWN
End algorithm
```

Algorithm flow: 1) Traverse the policies in the access control policy set and make the permission decision of access request AR. Three policy decision result sets, namely PERMIT_RESULT, DENY_RESULT, and UNKNOWN_RESULT, are obtained. 2) If only a single result, PERMIT or DENY, exists, the corresponding result (PERMIT or DENY) is returned. If the decision includes both PERMIT and DENY, or if the decision result is UNKNOWN, UNKNOWN is returned.

4.3 Business Permission Decision Algorithm Based on BiLSTM Network

(1) Algorithm concept

In BDAC-TPDS, the business permission decision algorithm based on the BiLSTM network is processed by the BDP module. The core concept of the algorithm is shown in Fig. 3. The business attribute sets of the subject and resource are expressed in the form of word vectors based on the Word2Vec model. The similarity between the business attribute sets of the subject and the resource is calculated based on a permission decision BiLSTM (PD-BiLSTM) neural network, and the decision result is output according to the similarity result.

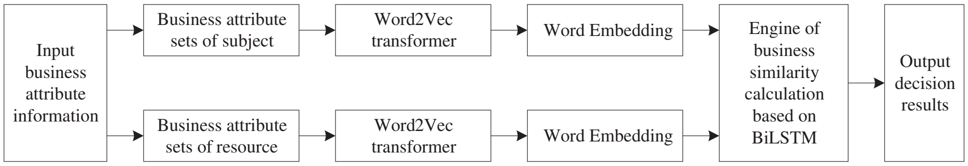

Input business attribute information
Business attribute sets of subject
Word2Vec transformer
Word Embedding
Business attribute sets of resource
Word2Vec transformer
Word Embedding
Engine of business similarity calculation based on BiLSTM
Output decision results

Figure 3: Algorithm concept

(2) Principles of neural network

A recurrent neural network (RNN) is a very powerful neural network owing to its internal storage structure. Each neuron can store previous input information using the storage structure. Therefore, an RNN can effectively process time series data. However, although the classical RNN can process time series information, it will lose its ability to learn past remote information with time. The most memorable information in the network is the last input information, which shows insufficient performance. LSTM (Long Short-Term Memory) has been proposed to overcome the existing problems in the RNN model. By introducing the memory cell structure, the neural network can realize long-term sequence information memory without the need for debugging complex super parameters. It has been widely used in many fields such as language modeling, machine translation, and speech recognition. Fig. 4 shows the memory cell structure in LSTM. Each cell structure contains four layers of the neural network. Because there may be correlations between attributes, an attribute may be associated with both its previous and next attributes. The LSTM can only use the historical data information, but cannot use the future data information in the data. Therefore, in this case, BiLSTM (Bi-directional Long Short-Term Memory) is used to link two LSTM with opposite timing directions into the same network output. This structure is shown in the BiLSTM Layer in Fig. 5. With this structure, BiLSTM adds computable information in LSTM, enabling the network model to obtain both historical and future information.

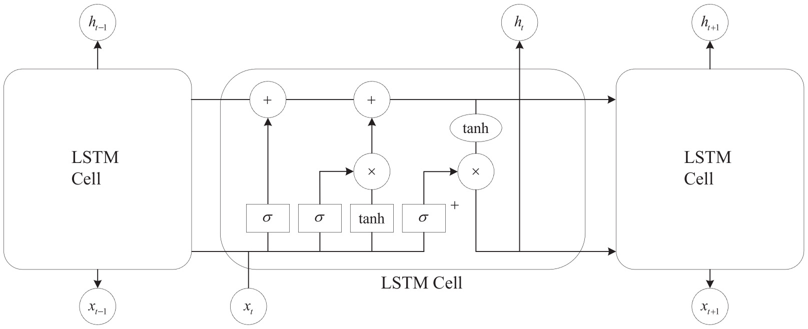


Figure 4: Memory cell structure of LSTM

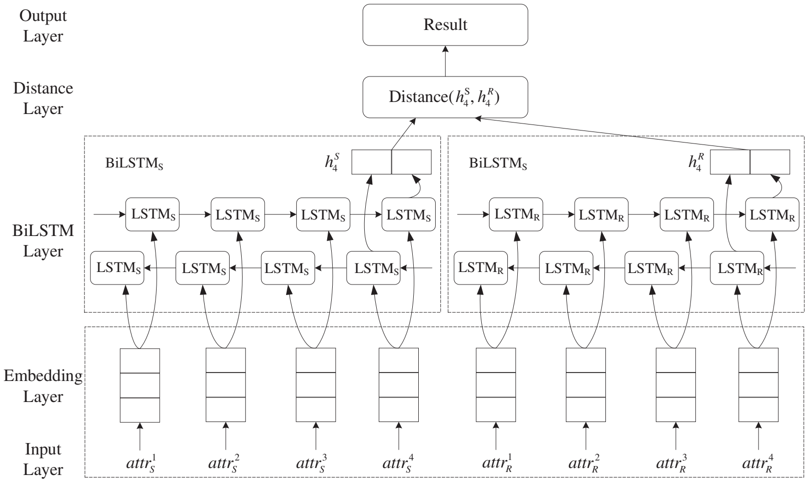


Figure 5: Neural network structure

LSTM uses the "gate" structure to delete and enhance the information stored in the memory cells. The core components of LSTM are the storage state C_t, output gate O_t, input gate it, and forget gate ft. Ot determines how the current storage state affects other memory cells. Based on the current state of the network, it and ft determine the information to be discarded and control the information input into the memory cells. The updating formulas of the LSTM network at time t are given below:

$$f_t = \sigma\left(W_f^x x_t + W_f^h h_{t-1} + b_f\right) \tag{4}$$

$$i_t = \sigma\left(W_i^x x_t + W_i^h h_{t-1} + b_i\right) \tag{5}$$

$$C_t^* = \tanh\left(W_c^x x_t + W_c^h h_{t-1} + b_c\right) \tag{6}$$

$$C_t = i_t \otimes C_t^* + f_t \otimes C_{t-1} \tag{7}$$

$$O_t = \sigma\left(W_O^x x_t + W_O^h h_{t-1} + b_O\right) \tag{8}$$

$$h_t = O_t \otimes \tanh\left(C_t\right) \tag{9}$$

where W_f^x, W_f^h, W_i^x, W_i^h, W_c^x, W_c^h, W_O^x, W_O^h is the weight matrix and b_f, b_i, b_c, b_O is the bias vector. hforward and hbackward are respectively hidden layer output vectors of BiLSTM forward LSTM and backward LSTM structure units. Link hforward and hbackward to obtain BiLSTM final output at time t. It is shown in Eq. (10).

$$C_t = concat\left(h_{forward}, h_{backward}\right) \tag{10}$$

(3) PD-BiLSTM model

The structure of the PD-BiLSTM model is shown in Fig. 5. PD-BiLSTM is a type of Siamese Network [28] with bidirectional LSTM structure. It is composed of two neural networks, namely BiLSTMS and BiLSTMR. BiLSTMS is used to process the business attributes of the subject, and BiLSTMR is used to process the business attributes of the resource. The input of each network is a set of business attributes. BiLSTMS and BiLSTMR have the same model weight parameters. PD-BiLSTM uses BiLSTM to read the word vector representing each business attribute and uses the final hidden state hfinal of the BiLSTM as the vector representation of each business attribute set. Then, the similarity between the final hidden states hfinal in the two networks is used for the prediction of business semantic similarity.

BiLSTM maps the set of business attributes of an entity from a vector space of a variable length sequence of length M to a constant multi-dimensional vector space VN of length N, where M is the dimension of the word vector and N is the maximum number of allowed attributes. In other words, the business attribute set of each entity is represented as the sequence {attr1, attr2, . . . , attrn}. The sequence is passed to BiLSTM, which updates the hidden state information at each sequence index by Eqs. (4)–(10). The final set of entity business attributes is encoded as the final hidden state hfinal∈VN of the model. For a given set of business attributes for the subject and resource, our model applies the predefined business permission decision function Verify: VN→VN→Result to the BiLSTM representation. The similarity of business attribute representation is used to infer the similarity of business attribute constraints between subjects and resources. Then, the business permission is decided by the similarity. The similarity calculation formula used in this study is given by:

$$\text{Distance}\left(h_{final}^{(S)}, h_{final}^{(R)}\right) = ||h_{final}^{(S)} - h_{final}^{(R)}||_2 \tag{11}$$

5 Security and Availability Analysis

(1) Privilege escalation attack

In a permission escalation attack, under normal operation of the access control mechanism, the attacker obtains higher unauthorized access permission to the system through a lower authorized permission. To implement such an attack in BDAC-TPDS, the attacker needs to obtain the business attributes of the resource Data1 through the access ability of the authorized resource Data1 after calculating the business permission similarity based on the BiLSTM network. Because there may be differences between Data1's business attributes and the attacker's business attributes, the attacker can access the unauthorized data resource Data2 through the business attributes of Data1. Thus, a permission escalation attack may be successful. However, this situation is not true in the BDAC-TPDS model. The user's business attribute is set by the security administrator when the user receives the corresponding business access demand. The user's business attribute cannot be changed through its access to resources; only the security administrator can change the user's business attribute. When the user's business is completed, the system can automatically withdraw the user's business attributes and cancel the user's access permission to relevant business resources. An attacker cannot change his own business attribute information; hence, he/she cannot access other business data. Thus, a permission escalation attack cannot be implemented in the BDAC-TPDS model.

(2) Attribute forgery attack

In an attribute forgery attack, the attacker can gain access to additional resources by forging entity attribute information. In BDAC-TPDS, the entity attribute information (including inherent attributes and business attributes) is uniformly managed by attribute certificates of the trusted attribute authority AA. In the process of parsing access requests, it is necessary to send attribute query requests to the attribute authority to ensure the correctness of the entity attributes. In the process of access control decision of business attributes, the validity of the attribute certificate issued by the attribute authority AA can be verified to avoid possible attacks on business attributes (such as attribute tampering, attribute reasoning, and attribute collusion). Changes in the subject's attribute information must be approved by the security administrator, and the entity's attribute information can only be generated by the security administrator or automated generation technology. Therefore, as long as the attribute authority is reliable, an attacker cannot forge the attribute.

(3) Security assurance

The BDAC-TPDS model manages users' permissions on the basis of the two-layer permission decision structure of the entity's inherent attributes and business attributes. The business attribute is not to extend the user's basic permission of inherent attributes but to narrow the user's permission for more granular and precise access control. Business attributes are used to restrict users to access only the resources related to the corresponding business in order to overcome the difficulty in dividing the permission granularity of the resource and the rigidity of user authorization under the condition of big data. Therefore, we believe that the introduction of business constraints further improves the security of the system on the basis of the classical ABAC model.

(4) Availability analysis

In BDAC-TPDS, the security administrator only needs to give the users a set of business attributes related to the user's business that can complete the authorization operation of the user. Meanwhile, the system can automatically determine whether the user can access the corresponding resources through the calculation of business similarity. This method is highly consistent with the characteristics of the need for effective sharing and utilization of massive resources in the big data environment, which overcomes the difficulty of effectively managing the permission of massive resources. It can effectively improve the efficiency of big data resource analysis and utilization under a specific business background and improve the availability of data resources to ensure their security.

6 Experimental Analysis

6.1 Experiment Settings

Simulation experiments were conducted on the Tensorflow1.12 platform to evaluate the effectiveness of the proposed method. The experimental software and hardware environment was as follows: Operating system, Windows 10 64-bit; CPU, Intel® Core™ i7-8750H @ 2.21 GHz; GPU, GeForce GTX 1050 Ti max-q; memory, 16 GB, Tensorflow version is 1.14.0, and the Keras version is 2.1.3. The simulation experiments were conducted on the basis of the text semantic similarity data set [29] published by the Stanford University SNLI project [30], which includes more than 360,000 experimental data marked with text similarity. In the simulation experiments, we randomly divided the data set into a training set of 300,000 data, a verification set of 30,000 data, and a test set of 30,000 data. In addition, we built access control policy sets with policy scales of 1000, 2000, 3000, 4000, 5000, 6000, 7000 and 8000 to conduct performance tests. These policy sets covered 100 subject attributes, 1000 resource attributes, and 10 action attributes. The maximum number of attribute data that describes the same subject, resource, and action was four. The maximum achievable distinguishable subject size space, resource size space, and action size space were 1004, 10004, and 104 based on the attribute information.

6.2 Experimental Results and Analysis

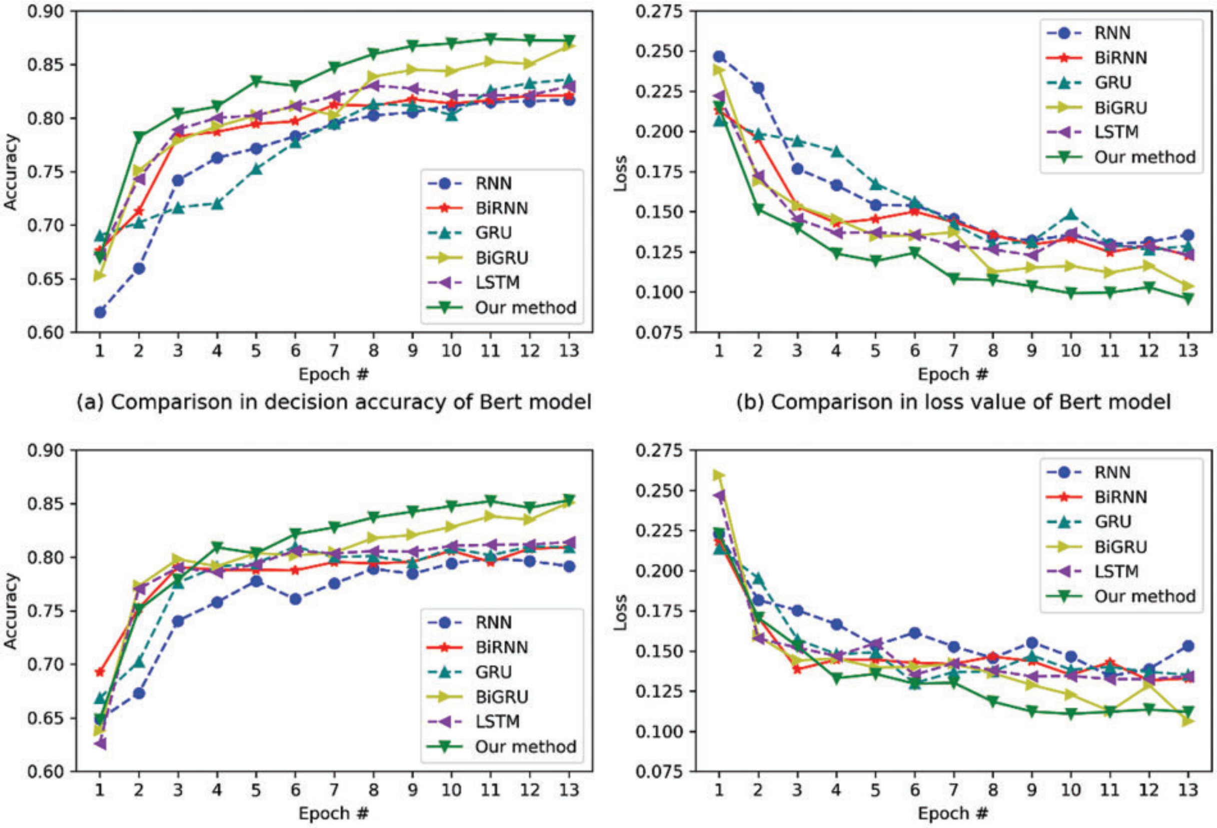
To evaluate the performance and effect of the proposed access control technology, we performed three experiments: A comparison of business permission decision performance among different methods, a comparison of calculation methods for different similarity distances, and a comparison of permissions decision time costs with benchmark methods.

(1) Comparison of business permission decision performance among different methods

To compare the performance of different neural network models in the process of business permission decision, five neural network models are selected as the benchmark for performance comparison. The baseline comparison models are RNN, BiRNN, GRU, BiGRU, and LSTM, respectively. The experiment adopts a control variable method to compare the performance of different methods under the condition of the same parameter. Meanwhile, we test the effect of different word vector models on decision performance. Figs. 6a and 6b use the latest Google pre-training language model BERT [31] (Bidirectional Encoder Representation from Transformers) as the Embedding Layer. Figs. 6c and 6d use the pre-training language model based on skip-gram as an Embedding Layer. The skip-gram pre-training model is based on Wikipedia data. The pre-trained language model is not the core of the paper. We use a publicly released pre-trained model.

In Fig. 6, the experiment results show that the performance of the business permission decision stabilizes after the 12th epoch. In the test data set, our method combined with the use of the Google BERT pre-training model can achieve optimal performance. The optimal accuracy rate reaches 87.39% and the optimal loss value reaches 0.0997. It can basically meet the requirements of access control under business permission decisions for big data. The performance of Google BERT pre-training model is better than the skip-gram pre-training model, which is caused by BERT's use of a Transformer network structure. It can implement more targeted training and more training layers. Since the pre-training language model of word vectors is not the focus of this paper, no more discussion will be given.

In Fig. 7, the experiment results show the influence of different hyper-parameters on the performance. Fig. 7a compares the effect of the number of neurons in the network on performance. When the number of neurons reaches 200, the performance is stable and will not continue to improve. Therefore, there is no need to use too many neurons in the network. Too many neurons will lead to overfitting of the model. Fig. 7b shows the influence of different sizes of batch_size on the accuracy rate during training. When batch_size reaches 150, the performance is basically stable, and the accuracy cannot be further improved by improving batch_size.



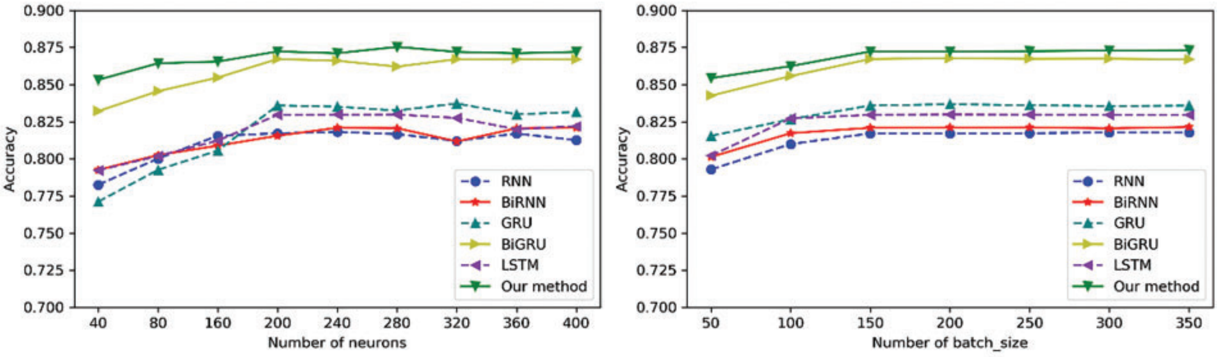
(a) Comparison in decision accuracy of Bert model

(b) Comparison in loss value of Bert model

(c) Comparison in decision accuracy of skip-gram model

(d) Comparison in loss value of skip-gram model

Figure 6: Performance comparison among different methods



(a) Comparison in the number of neurons

(b) Comparison in the number of batch_size

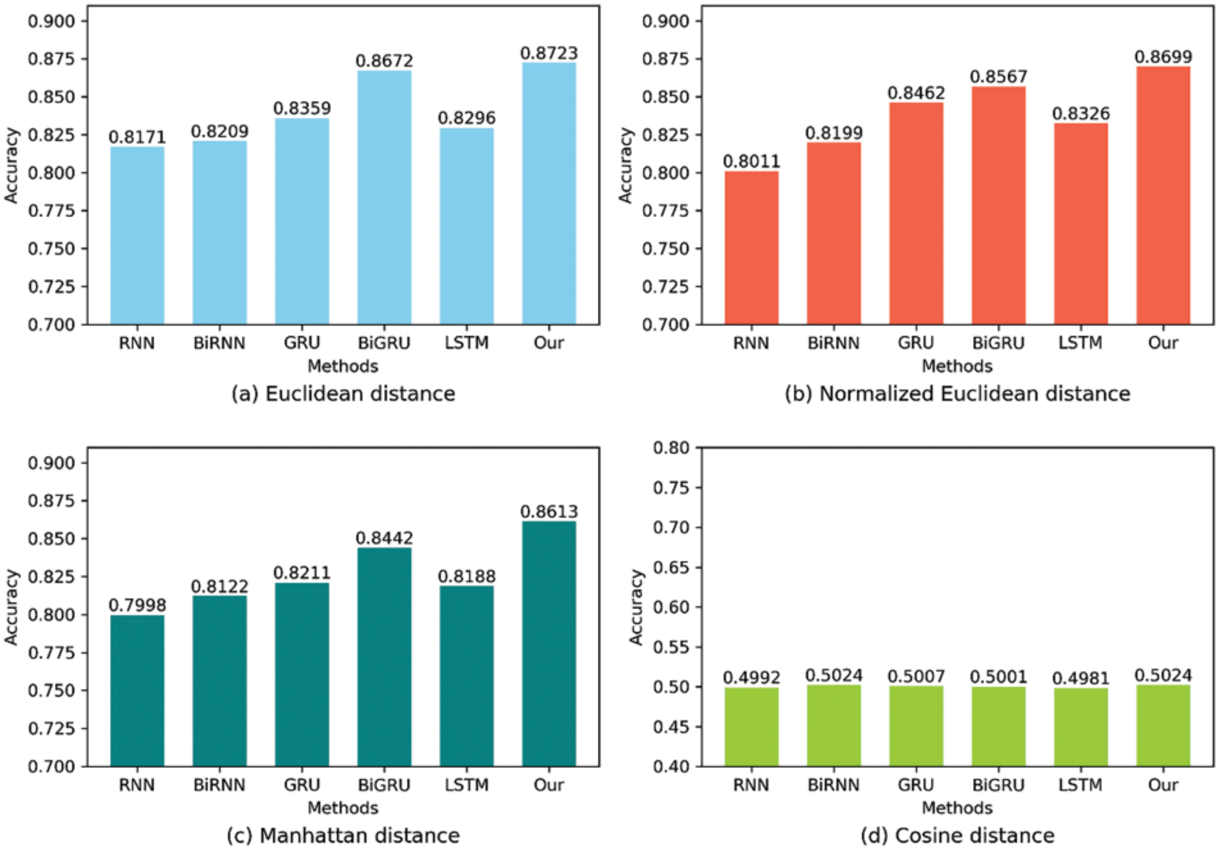
Figure 7: Effects of different hyper-parameters on performance

(2) Comparison of calculation methods for different similarity distances

Calculation methods for four similarity distances are shown in Table 3. As shown in Fig. 8 and 9, the calculation methods for four different business attribute similarity distances (Euclidean distance, Normalized Euclidean distance, Manhattan distance, and cosine distance) are compared in terms of decision performance.

Table 3: Calculation methods for four similarity distances

Number	Name	Calculation method
1	Euclidean distance	$\text{Distance}\left(h_{final}^{(S)}, h_{final}^{(R)}\right) = \lvert\lvert h_{final}^{(S)} - h_{final}^{(R)} \rvert\rvert_2$
2	Normalized Euclidean distance	$\text{Distance}\left(h_{final}^{(S)}, h_{final}^{(R)}\right) = \dfrac{\lvert\lvert h_{final}^{(S)} - h_{final}^{(R)} \rvert\rvert_2}{\lvert\lvert h_{final}^{(S)} \rvert\rvert_2 + \lvert\lvert h_{final}^{(R)} \rvert\rvert_2}$
3	Manhattan distance	$\text{Distance}\left(h_{final}^{(S)}, h_{final}^{(R)}\right) = \lvert\lvert h_{final}^{(S)} - h_{final}^{(R)} \rvert\rvert_1$
4	Cosine distance	$\text{Distance}\left(h_{final}^{(S)}, h_{final}^{(R)}\right) = \dfrac{\left(h_{final}^{(S)}\right)^T \cdot h_{final}^{(R)}}{\lvert\lvert h_{final}^{(S)} \rvert\rvert_2 + \lvert\lvert h_{final}^{(R)} \rvert\rvert_2}$



(a) Euclidean distance

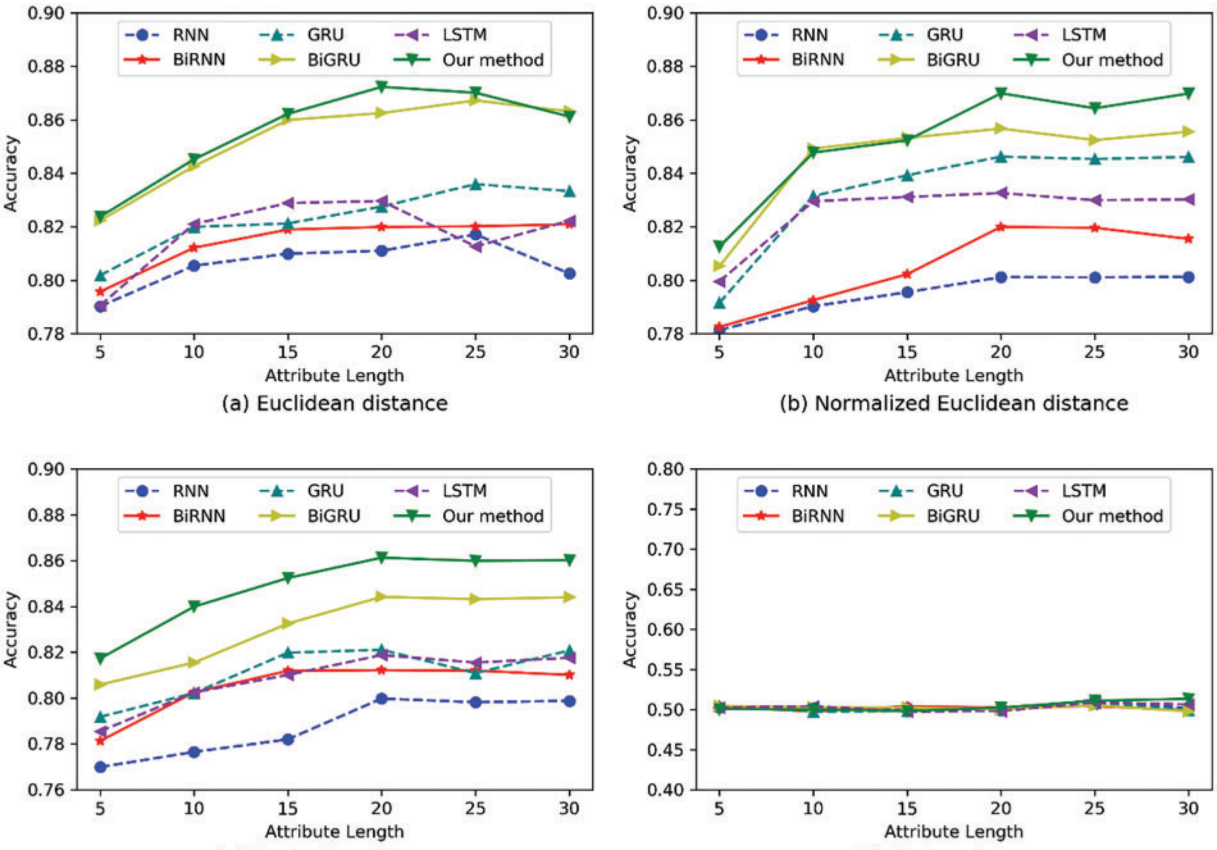
(b) Normalized Euclidean distance

(c) Manhattan distance

(d) Cosine distance

Figure 8: Comparison of decision accuracy for different similarity distances

It can be seen from the experimental results in Fig. 8 that the Euclidean distance, Normalized Euclidean distance, and Manhattan distance can achieve better decision effects. Among them, the actual decision effect that can be achieved by adopting the Euclidean distance is optimal, while the cosine distance adopted by the model cannot be fitted in the training process and its effect is the worst. In Fig. 9, we compare the effects of different attribute lengths on experimental performance. We can find that the model performs best when the attribute length is 20. This is because the attribute length is too small to mine the semantic information between attributes effectively. However, if the attribute length is too large, it will be difficult to realize the effective training of the model, and the attention will be more easily distracted during the training. Therefore, we choose to set the attribute length to be 20.



(a) Euclidean distance

(b) Normalized Euclidean distance

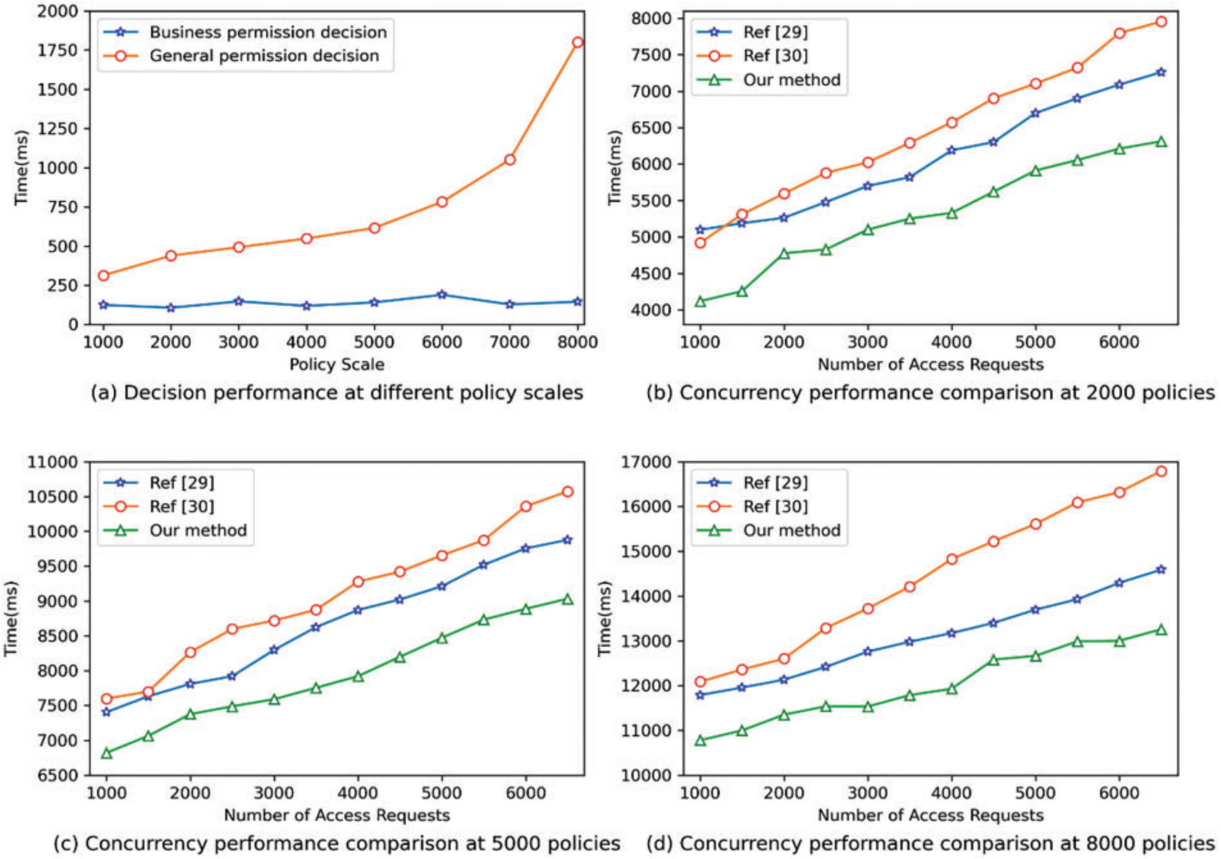
(c) Manhattan distance

(d) Cosine distance

Figure 9: Comparison of decision accuracy for different attribute lengths

(3) Comparison of permissions decision time costs with benchmark methods

Figs. 10a shows the required time costs for the general permission decision and business permission decision, respectively, under policy scales of 1000, 2000, 3000, 4000, 5000, 6000, 7000 and 8000. We can find that the time cost of the general permission decision increases with the policy scales. However, the time cost of the business permission decision is independent of the policy scales. This is mainly because the general permission decision requires logical calculation of the access request AR and all the policies in the policy set. Hence, an increase in the policy scales has a great impact on the delay of the general decision. The time complexity of the general permission decision is O(n), and n is the policy size. The time complexity of the business permission decision is O(1). This is because the implementation of the business permission decision only needs to input the business attribute information of the subject and resource into the decision engine based on the neural network to complete the decision. It does not need logical calculation with all the policies in the policy set. Hence, an increase in the policy scale does not have a significant impact on the delay of the business permission decision.



(a) Decision performance at different policy scales (b) Concurrency performance comparison at 2000 policies

(c) Concurrency performance comparison at 5000 policies (d) Concurrency performance comparison at 8000 policies

Figure 10: Performance comparison with benchmark methods

Meanwhile, in Figs. 10b–10d, we test the concurrent permission decision performance of different benchmark methods [32,33] under the 2000, 5000 and 8000 policy scales. Experimental results show that our method requires the least decision time under the same conditions. This is because business permission decisions based on similarity can significantly reduce the complexity of permission decisions. Therefore, this method is more suitable for big data scenarios with massive data resources.

7 Conclusion

Existing policy management methods based on manual formulation of the access control policy suffer from low efficiency and difficulty in accurately describing the access control policy. To overcome these problems, this paper proposed a big data access control mechanism based on a two-layer permission decision structure (BDAC-TPDS) as a new solution for automatic and intelligent big data access control, which can adapt to the business needs between entities. In addition, a two-layer permission decision structure was designed to account for the different access control requirements of big data resources. Furthermore, the security and availability of resources were also considered. Experimental results showed that the proposed method can achieve good permission decision results. However, there are some limitations to our method. The reliability of access control is also closely related to the accuracy of entity business attribute labeling. This paper focuses on the research of access control and does not consider the technology of business attribute annotation. In addition, to ensure the timeliness of processing massive and highly concurrent data access requests, performance optimization should be combined with big data systems in practical applications. In the future, we will also research high-performance business attribute annotation technology to provide attribute support for access control and optimize the performance of BiLSTM-based business permission decision algorithm to further improve decision efficiency and accuracy.

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

Funding Statement: This work is supported by Key Research and Development and Promotion Program of Henan Province (No. 222102210069), Zhongyuan Science and Technology Innovation Leading Talent Project (224200510003), 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, Na Wang, Xuehui Du; data collection: Aodi Liu, Dibin Shan; analysis and interpretation of results: Aodi Liu, Na Wang, Xiangyu Wu; draft manuscript preparation: Aodi Liu, Na Wang, Xuehui Du, Dibin Shan; validation: Wenjuan Wang, 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 SNLI project repository at https://nlp.stanford.edu/projects/snli/. The data that support the findings of this study are openly available at https://drive.google.com/open?id=1itu7IreU_SyUSdmTWydniGxW-JEGTjrv.

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

References

[1] J. L. Wang, C. Q. Xu, J. Zhang, and R. Zhong, "Big data analytics for intelligent manufacturing systems: A review," *J. Manuf. Syst.*, vol. 62, no. 2, pp. 738–752, Jan. 2022. doi: 10.1016/j.jmsy.2021.03.005.

[2] H. B. Abdalla, "A brief survey on big data: Technologies, terminologies and data-intensive applications," *J. Big Data-Ger.*, vol. 9, no. 1, pp. 1–36, Nov. 2022. doi: 10.1186/s40537-022-00659-3.

[3] C. L. Stergiou, K. E. Psannis, and B. B. Gupta, "InFeMo: Flexible big data management through a federated cloud system," *ACM Trans. Internet Technol.*, vol. 22, no. 2, pp. 1–22, May 2021. doi: 10.1145/3426972.

[4] A. P. Plageras, K. E. Psannis, C. Stergiou, H. Wang, and B. B. Gupta, "Efficient IoT-based sensor BIG Data collection-processing and analysis in smart buildings," *Future Gener. Comp. Sys.*, vol. 82, no. 1, pp. 349–357, May 2018. doi: 10.1016/j.future.2017.09.082.

[5] Q. Zhou, "A study on human transiting based on big data and web semantics: Distinguishment and detection," *Int. J. Semant. Web. Inf.*, vol. 18, no. 1, pp. 1–18, Feb. 2022. doi: 10.4018/IJSWIS.310055.

[6] L. Dong, T. Wu, W. Jia, B. Jiang, and X. Li, "Computable access control: Embedding access control rules into Euclidean space," *IEEE Trans. Syst. Man Cybern, Syst.*, vol. 53, no. 10, pp. 6530–6541, Oct. 2023. doi: 10.1109/TSMC.2023.3283527.

[7] J. Li, T. Wang, B. Yang, Q. Yang, W. Zhang, and K. Hong, "ABCrowdMed: A fine-grained worker selection scheme for crowdsourcing healthcare with privacy-preserving," *IEEE Trans. Serv. Comput.*, vol. 16, no. 5, pp. 3182–3195, Sep. 2023. doi: 10.1109/TSC.2023.3292498.

[8] D. Demiroll, R. Das, and D. Hanbay, “A key review on security and privacy of big data: Issues, challenges, and future research directions,” *Signal Image Video Process.*, vol. 16, no. 17, pp. 1335–1343, Sep. 2023. doi: 10.1007/s11760-022-02341-w.

[9] D. Shan, X. Du, W. Wang, N. Wang, and A. Liu, “KPI-HGNN: Key provenance identification based on a heterogeneous graph neural network for big data access control,” *Inform. Sci.*, vol. 659, no. 1, pp. 120059, Jan. 2024. doi: 10.1016/j.ins.2023.120059.

[10] R. Jiang, S. S. Han, Y. M. Yu, and W. P. Ding, “An access control model for medical big data based on clustering and risk,” *Inform. Sci.*, vol. 621, no. 1, pp. 691–707, Apr. 2023. doi: 10.1016/j.ins.2022.11.102.

[11] A. Almehmadi and K. El-Khatib, “On the possibility of insider threat prevention using intent-based access control (IBAC),” *IEEE Syst. J.*, vol. 11, no. 2, pp. 373–384, Jun. 2017. doi: 10.1109/JSYST.2015.2424677.

[12] J. Hong, K. Xue, Y. Xue, W. Chen, D. Wei, N. Yu *et al.*, “TAFC: Time and attribute factors combined access control for time-sensitive data in public cloud,” *IEEE Trans. Serv. Comput.*, vol. 13, no. 1, pp. 158–171, Jan. 2020. doi: 10.1109/TSC.2017.2682090.

[13] H. W. Liu, S. M. Meng, J. Hou, S. Wang, Q. M. Li, and C. Y. Huang, “Locality-sensitive hashing-based link prediction process on smart campus education or online social platform,” *J. Circuit Syst. Comp.*, vol. 31, no. 9, pp. 2250160, Jun. 2022. doi: 10.1142/S0218126622501602.

[14] L. Rikhtechi, V. Rafeh, and A. Rezakhani, “BBAC: Behavior-based access control to detect user suspicious behavior,” *J. Intell. Fuzzy Syst.*, vol. 43, no. 6, pp. 8207–8220, Nov. 2022. doi: 10.3233/JIFS-212377.

[15] H. Kim, D. K. Kim, and A. Alaerjan, “ABAC-based security model for DDS,” *IEEE Trans. Dependable and Secure Comput.*, vol. 19, no. 5, pp. 3113–3124, Sep. 2022. doi: 10.1109/TDSC.2021.3085475.

[16] M. Aftab *et al.*, “Traditional and hybrid access control models: A detailed survey,” *Secur. Commun. Netw.*, vol. 2022, no. 5, pp. 1560885, Feb. 2022. doi: 10.1155/2022/1560885.

[17] L. Zhao, M. Sun, B. B. Yang, J. P. Xie, and J. Q. Feng, “Zero trust access authorization and control of network boundary based on cloud sea big data fuzzy clustering,” *J. Intell. Fuzzy Syst.*, vol. 43, no. 3, pp. 3189–3201, Jul. 2022. doi: 10.3233/jifs-220128.

[18] W. Ben Daoud, M. Rekik, A. Meddeb-Makhlouf, F. Zarai, and S. Mahfoudhi, “SACP: Secure access control protocol,” in *Proc. IWCMC*, Harbin, China, 2021, pp. 935–941.

[19] R. Jiang, Y. Xin, H. P. Cheng, and W. X. Wu, “T-RBAC model based on two-dimensional dynamic trust evaluation under medical big data,” *Wirel. Commun. Mob. Comput.*, vol. 2021, pp. 1–17, Aug. 2021. doi: 10.1155/2021/9957214.

[20] R. Jiang, Y. Xin, Z. X. Chen, and Y. Zhang, “A medical big data access control model based on fuzzy trust prediction and regression analysis,” *Appl. Soft Comput.*, vol. 117, no. 11, pp. 108423, Mar. 2022. doi: 10.1016/j.asoc.2022.108423.

[21] L. Chen, X. Zhao, R. Zhao, G. Yuan, S. Zhang, S. Chen *et al.*, “Mobile internet access control strategy based on trust perception,” in *Proc. ICAIS*, Qinghai, China, 2022, pp. 598–611.

[22] R. Jiang, S. S. Han, Y. Zhang, T. W. Chen, and J. R. Song, “Medical big data access control model based on UPHFPR and evolutionary game,” *Alex. Eng. J.*, vol. 61, no. 12, pp. 10659–10675, Dec. 2022. doi: 10.1016/j.aej.2022.03.075.

[23] K. Ma and G. Yang, “RTBAC: A risk-aware topic-based access control model for text data with paragraph-level authorization,” *Secur. Commun. Netw.*, vol. 2022, no. 4, pp. 3371688, May 2022. doi: 10.1155/2022/3371688.

[24] M. Drozdowicz, M. Ganzha, and M. Paprzycki, “Semantic access control for privacy management of personal sensing in smart cities,” *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 1, pp. 199–210, Jan. 2022. doi: 10.1109/TETC.2020.2996974.

[25] Y. Verginadis *et al.*, “Context-aware policy enforcement for PaaS-enabled access control,” *IEEE Trans. Cloud Comput.*, vol. 10, no. 1, pp. 276–291, Jan. 2022. doi: 10.1109/TCC.2019.2927341.

[26] M. Alohaly, H. Takabi, and E. Blanco, “A deep learning approach for extracting attributes of ABAC policies,” in *Proc. SACMAT 2018*, New York, USA, 2018, pp. 137–148.

[27] A. Liu, X. Du, and N. Wang, "Unstructured text resource access control attribute mining technology based on convolutional neural network," *IEEE Access*, vol. 7, pp. 43031–43041, Mar. 2019. doi: 10.1109/ACCESS.2019.2907815.

[28] T. Muller, G. Perez-Torro, M. Franco-Salvador, and L. Assoc Computat, "Few-shot learning with Siamese networks and label tuning," in *Proc. ACL*, Dublin, Ireland, 2022, pp. 8532–8545.

[29] B. Samuel, A. Gabor, P. Christopher, and M. Christopher. DataSet, 2015. Accessed: Mar. 5, 2024. [Online]. Available: https://drive.google.com/open?id=1itu7IreU_SyUSdmTWydniGxW-JEGTjrv

[30] B. Samuel, A. Gabor, P. Christopher, and M. Christopher. SNLI Project, 2015. Accessed: Mar. 5, 2024. [Online]. Available: https://nlp.stanford.edu/projects/snli/

[31] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, Minnesota, USA, 2018, pp. 4171–4186.

[32] F. Deng *et al.*, "An efficient policy evaluation engine for XACML policy management," *Inform. Sci.*, vol. 547, no. 2021, pp. 1105–1121, Feb. 2021. doi: 10.1016/j.ins.2020.08.044.

[33] F. Deng, J. Lu, S. Y. Wang, J. Pan, and L. Y. Zhang, "A distributed PDP model based on spectral clustering for improving evaluation performance," *World Wide Web*, vol. 22, no. 4, pp. 1555–1576, Jul. 2019. doi: 10.1007/s11280-018-0588-8.