Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2024.047665

ARTICLE

Deep Learning Social Network Access Control Model Based on User Preferences

Fangfang Shan[1,2,*], Fuyang Li[1], Zhenyu Wang[1], Peiyu Ji[1], Mengyi Wang[1] and Huifang Sun[1]

[1]School of Computer Science, 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: 13 November 2023 Accepted: 25 January 2024 Published: 16 April 2024

ABSTRACT

A deep learning access control model based on user preferences is proposed to address the issue of personal privacy leakage in social networks. Firstly, social users and social data entities are extracted from the social network and used to construct homogeneous and heterogeneous graphs. Secondly, a graph neural network model is designed based on user daily social behavior and daily social data to simulate the dissemination and changes of user social preferences and user personal preferences in the social network. Then, high-order neighbor nodes, hidden neighbor nodes, displayed neighbor nodes, and social data nodes are used to update user nodes to expand the depth and breadth of user preferences. Finally, a multi-layer attention network is used to classify user nodes in the homogeneous graph into two classes: allow access and deny access. The fine-grained access control problem in social networks is transformed into a node classification problem in a graph neural network. The model is validated using a dataset and compared with other methods without losing generality. The model improved accuracy by 2.18% compared to the baseline method GraphSAGE, and improved F1 score by 1.45% compared to the baseline method, verifying the effectiveness of the model.

KEYWORDS

Graph neural networks; user preferences; access control; social network

1 Introduction

The rapid development of mobile communication technology and the progress of information sharing technology promote the wide application of social networks. According to the basic data of the 50th statistical report released by China Internet Network Information Center in June 2022, the number of Internet users in China is 1.051 billion, and the number of mobile Internet users is 1.047 billion. China's social networks have 893 million active users per day, with the average person spending 7.03 h a day on them. A large amount of data is generated on social network platforms every day.

People enjoy the convenience of social networks while also encountering privacy leakage issues. Sensitive information such as user identity, address, health status and financial account may be leaked, which may cause economic and reputation loss to users, and even threaten personal safety. Maintaining personal control over private information is one of the growing concerns of the digital society [1].

According to the Statistical Report on Internet Security Satisfaction of Chinese Netizens released by the Internet Society of China in 2020, 49.42% of netizens have infringed on personal information in their daily life, such as personal information leakage or excessive collection. Data has become a national production factor. Massive social network data is an important part of building a national unified data factor market, and social network access control can realize the safe and compliant use of data, which plays a core and key role in enabling and facilitating the healthy development of data factor market. How to ensure that data is controlled and shared in social networks and that authorized users access data according to regulations urgently needs to be solved.

As an important method to solve the problem of data security in social networks, access control technology not only denies unauthorized users' illegal access to data, but also guarantees that legitimate users can get access to data within the effective time. Many scholars have proposed many solutions to access control, but the following problems still exist. First, some existing work selects appropriate strategies to recommend to users according to the structure of social networks. This kind of work focuses more on the subject information in authorization and pays insufficient attention to the characteristics of objects. The access control model uses the same standard to develop user access policies, fails to integrate user preferences, lacks personalized services, and ignores the influence of user preferences on access control policies. Can not fully reflect the user preferences when the user authorization. Secondly, a common feature of access control models is the need to design abstract and intuitive access control policies. It is necessary to design access control information in the form of roles, attributes or relationships according to the actual situation, and then design access control rules. However, in a dynamic, complex, and large-scale social network environment, there are great limitations because it is difficult for administrators to maintain accurate access control state in the system [2].

This paper proposes a deep learning access control model based on user preference. According to the social network scene, the homogenous graph (user-user) and heterogeneous graph (user-social data) are constructed, and the graph neural network is used to simulate user social preferences and user personal preferences respectively (user preferences are formed by the integration of user social preferences and user personal preferences). User social preferences and user personal preferences are transferred and changed in the homogenous graph and heterogeneous graph respectively. Some users tend to be influenced by social neighbors and prefer user social preferences, while some users prefer to maintain their own style and prefer user personal preferences. When balancing the above two preferences, the model combines user social preferences and user personal preferences by weighting.

The main contributions of this paper are as follows:

1. The graph neural network is applied to access control to generate personalized access control policies that integrate user preferences. Traditional access control models extract information from access control states and simple access control rules to develop access control policies. However, during the extraction process, they fail to clearly observe user attributes and resources, resulting in poor generalization ability in the development of access control policies. The graph neural network model can solve the problem of poor generalization ability in access policy creation.
2. Two graph structures were constructed to simulate the propagation rules of user social preferences and user personal preferences in social networks, and the graph nodes were updated adaptively. A hidden neighbor node selection module is designed to obtain hidden nodes (nodes with no edge connection to the current node but within a specified similarity range), and a K-order neighbor node selection module is designed to obtain K-order neighbor nodes. When

updating graph nodes, not only displayed neighbor nodes (nodes with edge connections to the current node) and social data nodes are considered, but also hidden neighbor nodes and K-order neighbor nodes, which expand the scope and depth of user preferences.We fine-tuned the ChatGLM-6B model and performed information extraction on the data we crawled from the internet, resulting in structured data that facilitates subsequent model training.

3. When updating nodes in homogeneous and heterogeneous graphs, the attention mechanism is used to pay dynamic attention to local features. Weighted aggregation of user nodes in homogeneous and heterogeneous graphs is carried out to integrate user social preferences and user personal preferences to obtain user preferences, so that the model can obtain better node representation in sparse heterogeneous social networks and ensure the optimal performance of the model.
4. The experimental results show that the model proposed in this paper can ensure the correctness of the generated access control policies, and the accuracy is improved by 2.18% compared with the baseline model, and the model performance is better than the baseline model.

The rest of this article is organized as follows. Section 2 discusses some mainstream access control models and their drawbacks. Section 3 discusses the model architecture and overall execution process. Section 4 conducts experimental verification of the model and compares it with other baseline models. Finally, Section 5 concludes the paper and outlines future research plans.

2 Related Work

In recent years, many scholars have studied access control models for social networks and have achieved quite rich results. Based on different classification criteria, they can be divided into access control models based on trust, rules, attributes, and other methods.

2.1 Trust-Based Access Control Model

In both real life and social networks, the relationship between people is built on the foundation of trust. An essential prerequisite for a person to decide to interact with others is that they can establish enough trust with them [3]. Voloch et al. [4] proposed a trust-based access control model. This model classifies the direct connections between users into roles according to the user network. Meanwhile, it relies on the total number of friends, user account age, friendship duration and other public information to represent the network connection quality, and is used to evaluate the degree of trust between users and network members, so as to achieve access control. Wang et al. [5] proposed a trust-based access management framework (TBAF), that incorporates privacy awareness and access control. Access control modeling is carried out according to the relationship between individuals, sticky strategies in social networks, fine-grained format and degree of trust. The model supports the generation of highly complex privacy-related policies, so as to support privacy-protecting access control.

In view of the system complexity faced by cloud computing, Aparna et al. [6] proposed a trust-RBAC access control model, which integrates non-security oriented Trust of role-based access control (RBAC). Before granting authorization to the requested data unit, the user trust value is considered to improve the effectiveness of the authentication and authorization process in cloud computing. This provides a new dimension to prevent unauthorized access to cloud resources. Susan et al. [7] proposed an initial trust authorization access control model for unknown users in social networks based on the existence of relationships between them. A trust network is created based on existing relationships in the social network to evaluate the trustworthiness of users towards resources, and thereby restrict access to them. Xu et al. [8] proposed a trust-based access control mechanism Trust2Privacy, which defines trust directions between users based on their attention states and calculates trust values by considering the similarities, correlations, and interactions between users. The model protects user-generated social data and effectively transforms trust calculations into privacy protection. However, the calculation of trust values in the trust-based access control model mainly relies on the trust model, and the consideration of user attribute information and authorization context environment was lacking in the calculation process.

2.2 Rule-Based Access Control Model

This type of access control primarily operates on a predefined set of access management policies. If certain policies or rules within this set are satisfied, the user can access specific resources [9]. To address privacy and security issues related to the sharing of jointly owned data in social networks, Ma et al. [10] proposed a privacy protection policy rule fusion method. The method defines the scope of information protection based on the content of privacy data, abstracts natural language descriptions of privacy protection using predicate logic formulas, constructs a logical model for privacy protection rules, defines heterogeneous privacy protection rules, and provides a rule fusion algorithm to restrict data access permissions, thereby avoiding conflicts in privacy protection requirements and privacy data leakage in different application scenarios of social networks. Masoumzadeh et al. [11] proposed a theory to capture the semantics of rule-based access control policies. The method captures the semantics of access control policies in a formal manner, has a simple representation, and allows for intuitive policy descriptions, thereby enabling better selection of access control policies.

Dundua et al. [12] developed a rule-based unified framework for specifying and analyzing entity attributes in social networks, and allocated specific text control policies based on these attributes. Ma et al. [13] proposed a distributed logical social network access control model (DUD_RuleSN), based on active rules and trigger mechanisms, which incorporates relationships between entities into authorization, providing the model with strong authorization expression capabilities and updateability. Marin et al. [14] proposed a rule-based normative configuration and programming system semantic framework, which transforms norms into attribute-based access control executable code for access control. However, the biggest disadvantage of rule-based access control is that it cannot satisfy everyone's interests, as different people have different personal preferences. Therefore, personalized access control mechanisms become increasingly important. At the same time, rule-based access control lacks flexibility. If it is necessary to adjust new resources or access control policies, they must be modified in the access management policy set, which may cause conflicts or result in access being denied.

2.3 Attribute Based Access Control Model

Attribute-based access control is a classic type of access control model and also a type of access control technology suitable for open environments. Defining authorization with security attributes can effectively protect users' privacy information. Karimi et al. [15] proposed a method for automatically learning ABAC policy rules from system access logs, using an unsupervised learning algorithm to detect patterns in access logs and extract ABAC authorization rules from them. Two improved algorithms for rule pruning and policy refinement were also proposed to generate higher-quality mining strategies. Yahiatene et al. [16] proposed a framework called CloudSN to mitigate privacy issues in OSNs. Using a distributed multi-authority ABE scheme, it provides flexible access to private data, which can only be accessed by users with the correct keys. Wei et al. [17] proposed an intelligent privacy protection method to solve privacy protection issues in social networks, combining neural networks with a hybrid hierarchical genetic algorithm to construct a social network security prediction model. The social network information was preprocessed using SVM and encrypted using an attribute-based encryption scheme, and finally, the particle swarm optimization algorithm was used to improve the security and privacy protection of the social network.

Li et al. [18] proposed a privacy-aware ABE scheme with an accountability mechanism for multi-authority ciphertext policies, hiding attribute information in ciphertext and allowing tracking of the identities of dishonest users who share decryption keys. It was applied to cloud computing to construct a reliable fine-grained access control system. Safi et al. [19] proposed a fully encrypted privacy protection scheme based on mobile social networks, using ciphertext policy attribute-based encryption (CP-ABE) and advanced encryption standard (AES) to encrypt user data end-to-end. In the CP-ABE encryption scheme, access control is applied to other users' access to shared data in addition to data confidentiality, and unauthorized users cannot access or violate data privacy. However, when facing users with multiple attributes, attribute-based access control methods need to consider attribute weights, and as the number of user attributes increases, the algorithm's processing speed will slow down, and the response time for users will increase.

2.4 Other Access Control Models

In response to the shortcomings of large granularity and poor flexibility in access control policies, Hou et al. [20] proposed a multi-layer access control mechanism based on blockchain called BMAC. They designed an algorithm based on InfoMap to implement user grouping based on credibility and combined it with a multi-blockchain to establish a flexible and fine-grained trusted data access control mechanism. Ma et al. [21] proposed a content-based access control model (RCBAC) that is risk-aware. RCBAC uses the user's responsibilities and the content of the requested file to determine whether to grant access, and also divides user risks into access behavior risks and access history risks, dynamically adjusting the user's access capabilities through risk quantification and management. Li et al. [22] proposed a control framework called PrivacyJPEG for privacy protection of photo sharing across different social networks. They first encrypt several privacy areas of the photos and bind the access control policies before users upload the photos, which can limit any user on the same transmission chain from accessing any privacy area of the photo, even if they are in different social networks and not affected by unwanted viewers. Sun et al. [23] proposed a blockchain-based encrypted gradient auditing method to defend against attacks. It utilizes a behavior chain to record the encrypted gradients of data owners and employs an auditing chain to evaluate the quality of the gradients. This method adopts a privacy-preserving homomorphic noise mechanism where the noise of each gradient gets canceled out after aggregation, ensuring the usability of the aggregated gradients.

Yin et al. [24] designed the training process of federated learning (FL) as a game model. They utilize an extensive game tree to analyze the key elements influencing decision-making in a single game and then find incentive mechanisms that align with social norms through repeated games. Through multiple rounds of games, the incentive mechanism can assist all participants in finding the optimal strategies for energy, privacy, and accuracy in FL within a distributed communication system. Akkuzu et al. [25] proposed a group decision-making model based on consensus. When a user wants to publish data that is collectively owned, they first ask for group decisions from all owners during the sharing process, and then make the final decision by respecting or disrespecting the group decision. Sharma et al. [26] proposed a blockchain based framework with encryption based on ciphertext policy attributes for access control and user revocation. Shan et al. [27] proposed a social network forwarding control mechanism based on game theory by analyzing the interests of both forwarding parties. They compared the historical data of forwarding operations with the threshold set by the publisher to determine whether to allow forwarding. Hu et al. [28] proposed group-based access control (oGBAC) to ensure data security when sharing information between friends in OSN by limiting the information flow between groups. It is used to prevent privacy leaks when sharing information within or between groups in online social networks. However, all the above access control models ignore the participation of user preferences in the access control policy when granting social access control.

3 Model Structure

In this section, we first introduce some relevant definitions of homogeneous and heterogeneous graphs, and then describe an execution process of the entire model.

3.1 Related Definitions

There are two groups of entities in social network, which are user set $U(|U| = m)$ and social data set $D(|D| = z)$. Two graphs are constructed based on the user set U and the social data set D one is the user-user undirected heterogeneous graph $G_S = (U, S)$, and the other is the user-social data undirected homogenous graph $G_D = (U \vee D, C)$. The user-user social relationship matrix $S \in R^{m \times n}$ represents the social relationship between social users in the social network. User node a is represented by u_a and social data node i is represented by d_i. If u_a and u_b are friends, then $S_{ab} = S_{ba} = 1(b \in S_a)$, otherwise $S_{ab} = S_{ba} = 0(b \notin S_a)$. $Y \in R^{m \times z}$ represents the user-social data matrix. If u_a is associated with d_i, $C_{ai} = C_{ia} = 1$, otherwise $C_{ai} = C_{ia} = 0$. The feature vector matrix of all user nodes is $P \in R^{m \times h_1}$, and the feature vector of u_a is represented by p_a. The feature vector matrix of all social data nodes is $Q \in R^{z \times h_2}$, and the feature vector of d_i is q_i.

3.2 Problem Description

This paper proposes a deep learning access control model based on user preferences and simulates user preferences using graph neural networks. When user u_a generates a social data d_i, the model generates appropriate access control policies for the social data d_i based on user preferences.

As shown in Fig. 1, the homogenous graph and heterogeneous graph were constructed, and the graph nodes were embedded to obtain feature vectors as input. After the training and updating of the graph neural network and linear neural network, the user node in the user-user graph is classified and output. The classified category is the user's operation authority on the social data.

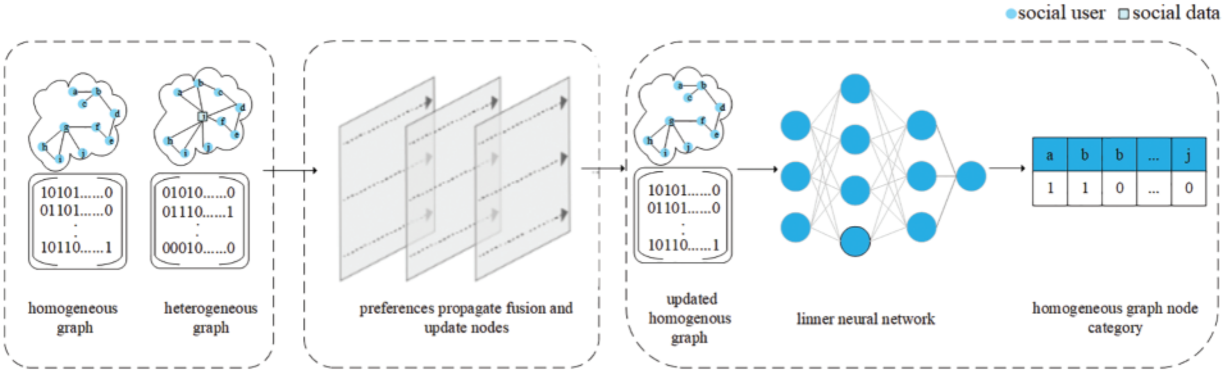

Figure 1: Model architecture execution flow chart

Specifically, user u_j has two possible access rights to the social data d_i generated by user u_a. If u_j can access social data d_i, the node category of u_j in the homogeneous graph is 1; otherwise, the node category of u_j is 0. The access control problem of social data is transformed into the classification problem of user nodes in the homogenous graph.

3.3 Detailed Structure of the Model

The model consists of three parts: embedding layer, preference propagation fusion layer and access control output layer.

The embedding layer embeds social users and social data to obtain feature vectors, which are taken as the input of the model.

In the preference propagation fusion layer, there are homogenous graph and heterogeneous graph, which respectively simulate the propagation rules of user social preferences among social users and the propagation rules of user personal preferences in social data. According to displayed neighbor node, hidden neighbor node, social data node and high-order neighbor node, the attention mechanism is used to update the user node. The attention mechanism is used to update the social data node according to the user node.

In the access control output layer, the linear neural network is used to calculate the fusion coefficient, and the user nodes obtained after multiple preference propagation and fusion are aggregated, and the final representation of the user nodes with user preferences is obtained in the homogenous graph. After activation function processing, the final node of each user node in the homogenous graph is dichotomized (0 is to deny the user access, 1 is to allow the user access), and the access control list is output. The model structure is shown in Fig. 2.

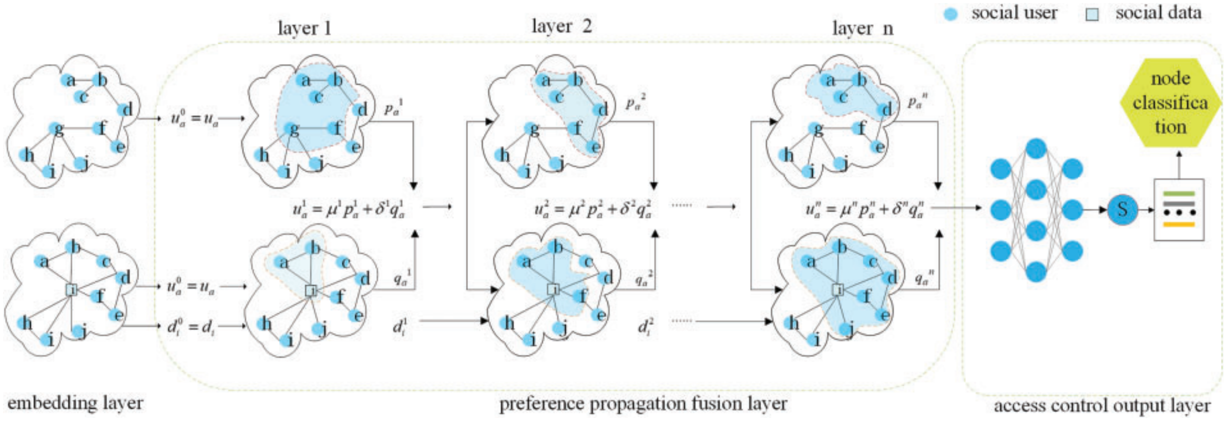
social user
social data
layer 1
layer 2
layer n
$u_a^0 = u_a$
$u_a^0 = u_a$
$d_i^0 = d_i$
p_a^1
p_a^2
p_a^n
q_a^1
q_a^2
q_a^n
$u_a^1 = \mu^1 p_a^1 + \delta^1 q_a^1$
$u_a^2 = \mu^2 p_a^2 + \delta^2 q_a^2$
$u_a^n = \mu^n p_a^n + \delta^n q_a^n$
d_i^1
d_i^2
S
node classifica tion
embedding layer
preference propagation fusion layer
access control output layer

Figure 2: Structure diagram of model detail

3.3.1 Embedding Layer

In the paper, during the dataset preparation phase, we fine-tuned the ChatGLM-6B large model with the aim of performing information extraction from the raw dataset and extracting key information from user attributes to represent them as user nodes. Fine-tuning is a common technique that involves further training a pre-trained model to better adapt it to specific tasks and datasets.

Firstly, the ChatGLM-6B model has learned rich feature representations and possesses strong generalization capabilities. Next, we fine-tuned ChatGLM-6B on our own crawled dataset using P-tuning v2 [29], with the default parameters specified in the fine-tuning scheme. By fine-tuning on the original dataset, the model's parameters were adjusted to a state that is more suitable for the current task.

The following is an example of information extraction from a dataset:

"I am an author. I enjoy exploring the world and chatting about everyday life. I am a sincere person who always finds a place to chat, even if it's just casual conversation. I was born on July 30, 1997, under the Zodiac sign Leo. I am located in Beijing and currently studying at Beijing Film Academy. I joined Weibo on March 12, 2011."

{ "Username": "Not provided",

"Bio": "Explore the world, chat about daily life. Just Hu chatting and listening. A sincere person always finds a place to chat and have random conversations.",

"Birthday": "1997-07-30",

"Other Information": {

"Zodiac Sign": "Leo",

"City": "Beijing",

"University": "Beijing Film Academy",

"Join Date on Weibo": "2011-03-12"} }

Selecting user attributes to represent user nodes in the graph, and obtain the feature vector matrix $P \in R^{\mathrm{m}\times h_1}$ of user nodes after embedding. Using CoreNLP to process and extract each social data item, and obtain the feature vector matrix $Q \in R^{z\times h_2}$ of social data nodes after embedding. Create free feature vector matrix $T \in R^{m\times d_1}$ of user node and free eigenvector matrix $V \in R^{z\times d_2}$ of social data node. The feature vector of a single user node is $u_a = f(W_1 \times [p_a, t_a]), p_a \in P, t_a \in T, u_a \in R^h$. The feature vector of a single social data node is $d_i = f(W_2 \times [q_i, v_i]), q_i \in Q, v_i \in V, d_i \in R^h$. Concatenate the free feature vector matrix for social users and social data to prevent the omission of important information during embedding construction, which could affect the overall model results.

3.3.2 Preference Propagation Fusion Layer

In this paper, we take the update of a single user node and social data node as an example. The user node feature vector u_a and the social data node feature vector d_i in the embedding layer serve as the input of the preference propagation fusion layer, which recursively simulates the propagation and diffusion of user social preferences and user personal preferences in the graph G_S and graph G_D. The initial feature vector of user node in graph G_S and graph G_D is $u_a^0 = u_a$, and the initial feature vector of social data node in graph is $d_i^0 = d_i$. In the n + 1 layer, update u_a^n and d_i^n in the n layer to obtain u_a^{n+1} and d_i^{n+1}. N is an important recursive parameter in model training. When the n = N recursion is carried out, the preference propagation fusion ends. The following section describes the updating process of user nodes and social data nodes in detail.

(1) Social user node update

User nodes are updated based on their neighbor nodes. When updating users nodes in graph G_S, two key issues need to be considered. Node embedding effectiveness plays a decisive role in model performance. The quality of node embedding effectiveness is closely related to the choice of neighbor nodes. Firstly, when updating user nodes, not only displayed neighbor nodes but also hidden neighbor nodes should be considered. Expanding the selection range of neighbor nodes actually increases the breadth of user preferences. Secondly, when updating user nodes, considering the K-order neighbors nodes actually expands the depth of user preferences.

Displayed neighbor nodes do not need to be selected. For hidden neighbor node selection, a hidden neighbor node selection module is designed to calculate the similarity score between two nodes. The similarity score is calculated as shown in Eq. (1). If the similarity of the two nodes approaches 1, it indicates that the two nodes are extremely similar. In this case, selecting this node as a hidden neighbor node to update the user node will cause information redundancy. If the similarity between two nodes approaches 0, it indicates that the information gap between two nodes is particularly large. At this time, selecting this node as a hidden neighbor node to update the user node will generate noise. It is very important to find an appropriate similarity score threshold for hidden node selection. After many experiments, the hidden neighbor node selected when the threshold is set as (0.5, 0.7) has the best effect on node updating.

$$sim\_more(u_a^n, u_b^n) = \|u_a^n\| \|u_b^n\| \cos\theta, S_{ab} = S_{ba} = \mathbf{0} \tag{1}$$

In Eq. (1), u_a^n and u_b^n represent the feature vectors obtained after the n-th embedding of nodes u_a and u_b. $b \notin S_a$ ($S_{ab} = S_{ba} = 0$) indicates the node that has no relationship with u_a^n. If u_b^n is selected as the hidden node, the relationship value of u_b^n and u_a^n in the user-user social relationship matrix is set to 1, that is $S_{ab} = S_{ba} = 1$.

The K-order neighbor node selection module is set. When the user node is updated, the K-order neighbor node of the user node is considered. A large K value results in excessive aggregation of redundant information. A small K value results in insufficient aggregation information. For the choice of K value, after many experiments, the value of K is set to 3.

After completing the hidden neighbor nodes and K-order neighbor node selection, the user node is updated. In graph G_S, the feature vector of the temporary node after the (n + 1)th propagation and update of u_a^n is p_a^{n+1}. The updating method is shown in Eq. (2). α_b^n represents the aggregate weight of each neighbor node when u_a^n is updated according to its neighbor node. The calculation method of α_b^n is shown in Eq. (3).

$$p_a^{n+1} = u_a^n + Agg(\{u_b^n \mid b \in S_a\}) = u_a^n + \sum_{b \in S_a} \alpha_b^n u_b^n \tag{2}$$

$$\alpha_b^n = softmax(\mathrm{R}elu(u_a^n)^T W_3 \mathrm{R}elu(u_b^n)), b \in S_a \tag{3}$$

In graph G_D, the temporary feature vector of the node after the (n + 1)th propagation and update of u_a^n is q_a^{n+1}. The updating process is shown in Eq. (4).

$$q_a^{n+1} = u_a^n + \mathrm{Agg}(\{d_i^n \mid i \in C_a\}) = u_a^n + \sum_{i \in C_a} \beta_i^n d_i^n \tag{4}$$

$$\beta_i^n = soft\max(MLP_1[u_a^n, d_i^n]), i \in C_a \tag{5}$$

In Eq. (4), $i \in C_a$ represents all social data nodes associated with u_a^n in graph G_D. β_i^n represents the weight of each social data node in the updating process of u_a^n. The calculation method of β_i^n is shown in Eq. (5). The feature vector of user nodes and social data nodes have different meanings in each dimension, so the neural network is used to calculate the aggregate weight of each social data node. Finally, the results obtained by the neural network are normalized to obtain β_i^n.

p_a^{n+1} from G_S and q_a^{n+1} from G_D are aggregated weighted to obtain user node feature vector u_a^{n+1} with user preferences, as shown in Eq. (6). μ^{n+1} and δ^{n+1} represent the aggregate weights of p_a^{n+1} and q_a^{n+1} when they update u_a^{n+1} together.

$$u_a^{n+1} = \mu^{n+1} p_a^{n+1} + \delta^{n+1} q_a^{n+1}, \mu^{n+1} + \delta^{n+1} = 1 \tag{6}$$

(2) Social data node update

In graph G_D, there are two types of input data, namely u_a^n and d_i^n. The updating process of the feature vector of social data for the n + 1 time is shown in Eq. (7), and the feature vector after social data updating is denoted as d_i^{n+1}.

$$d_i^{n+1} = d_i^n + Agg(\{u_a^n \mid a \in C_i\}) = d_i^n + \sum_{a \in c_i} \gamma_a^n u_a^n \tag{7}$$

$$\gamma_a^n = MLP_2[d_i^n, u_a^n], a \in C_i \tag{8}$$

In Eq. (7), $a \in C_i$ represents all user node associated with d_i^n in graph G_D. γ_a^n represents the aggregate weight of each neighbor node when d_i^n is updated. The calculation method of γ_a^n is shown in Eq. (8). Neural network is used to calculate the aggregate weight of each neighbor node, and the results obtained by each neural network are normalized to get the coefficient γ_a^n. After N times of propagation and fusion, the final user node represents $\{u_a^1, u_a^2, \ldots, u_a^N\}$.

3.3.3 Access Control Output Layer

After N propagations and fusions, there are N representations of u_a, denoted by $\{u_a^1, u_a^2, \ldots, u_a^N\}$. Calculate the fusion coefficients according to Eq. (9).

$$C_{u_a^n} = Soft\max(\tanh(W_4 u_a^n + e) \cdot Att^T) \tag{9}$$

Att is the attention feature vector used for integrated semantics, and W4 and e are the weights and biases of the fully connected layer, which projects the semantic representations of different nodes into the same vector space. The fusion coefficients $C_{u_a^n}$ are obtained using softmax normalization. The final user node is calculated based on the fusion coefficients and classified using the sigmoid function, as shown in Eq. (10).

$$u_a = sigmod\left(\sum_{n=1}^{N} C_{u_a^n} \mathrm{u}_a^n\right) \tag{10}$$

Define the loss function as shown in Eq. (11). y_a indicates the category of node u_a. The positive class is 1 (access allowed), and the negative class is 0 (access denied). pr_a represents the probability that the category of node u_a is 1. Based on the back propagation and gradient descent training model parameters, all node categories (1 or 0) are finally obtained.

$$L = \frac{1}{M} \sum_a - [y_a \bullet \log(pr_a) + (1 - y_a) \bullet \log(1 - pr_a)] \tag{11}$$

The overall execution process of the model is shown in Algorithm 1.

Algorithm 1 : Model execution process

```
Input: $G_S$,$G_D$, $P \in R^{m\times h_1}$, $Q \in R^{z\times h_2}$, N, K, W1, W2, W3, W4, MLP1, MLP2, S.
Output: categories of all user nodes $u_a$
#Input of graph node data
$u_a = f(W_1 \times [p_a, t_a]), p_a \in P, t_a \in T, u_a \in R^h$
$d_i = f(W_2 \times [q_i, v_i]), q_i \in Q, v_i \in V, d_i \in R^h$
# Compute the hidden nodes that are within the similarity range of user node a, and set the social
  matrix value between them to 1.
for $b \in S$ do
    if $sim\_score(u_a, u_b) in (0.5, 0.7) and S_{ab} = S_{ba} = 0 and order(u_a, u_b) = K$ then
        $S_{ab} = S_{ba} = 1$
    end if
    $u_a^0 \leftarrow u_a$
   $d_i^0 \leftarrow d_i$
end for
for $n = 0...N$ do
   # Update the user nodes
   $p_a^{n+1} = u_a^n + \sum_{b\in S_a} \alpha_b^n u_b^n$
   $\alpha_b^n = soft\max(\mathrm{Re}lu(u_a^n)^T W_3 \mathrm{Re}lu(u_b^n)), b \in S_a$
   $q_a^{n+1} = u_a^n + \sum_{i\in C_a} \beta_i^n d_i^n$
   $\beta_i^n = soft\max(MLP_1[u_a^n, d_i^n]), i \in C_a$
   $u_a^{n+1} = \mu^{n+1} p_a^{n+1} + \delta^{n+1} q_a^{n+1}$
   # Update the social data nodes
   $d_i^{n+1} = d_i^n + \sum_{a\in c_i} \gamma_a^n u_a^n$
   $\gamma_a^n = MLP_2[d_i^n, u_a^n], a \in C_i$
end for
for $n = 1...N$ do
    # compute fusion coefficient
    $C_{u_a^n} = Soft\max(\tanh(W_4 u_a^n + e) \cdot Att^T)$
end for
$u_a = sig\mathrm{mod}(\sum_{n=1}^{N} C_{u_a^n} \mathrm{u}_a^n)$
return $u_a$
```

4 Experiment

In this section, the model is compared with benchmark methods on a custom dataset to validate the effectiveness of the proposed model.

4.1 Dataset

The experiment used a custom dataset collected from the microblog social platform. Climb 50 users in the microblog social platform, each user has an average of 50 social friends, and 50 users climb to the 3 order social friends. Each user generates 20 pieces of social data, and each piece of social datasets access control policies for friends (allow access to this piece of social data or do not allow access to this piece of social data, because the social data is basically allowed to access other friends, after data collection, the experimenters in this group manually adjusted the data access control strategy). Based on each user's interaction with their social friends and the historical access control policy set for each piece of data, a complete dataset is finally constructed. The dataset details are shown in Table 1.

Table 1: Dataset

Number of user nodes	Number of social data nodes	Number of edge
10,081	1,000	60,195

4.2 Experimental Setup

This paper is based on the PyTorch framework to implement the model. The preference propagation depth (number of iterations) N of the model is set to 5. K in the K-order neighbor node selection module is set to 3. The model used SGD optimizer and the learning rate was set to 0.001. The early stop strategy is adopted during training. If the loss of verification set does not change during 100 consecutive iterations, the model training will be terminated.

4.3 Evaluation Index

In deep learning, binary classification problems commonly used important evaluation metrics are accuracy (Accuracy) and F1 score. They are calculated based on the confusion matrix. In experiments, we selected Accuracy and F1 score as indicators to evaluate the performance of the model. The meaning of accuracy mainly refers to the ratio of the number of samples that the model predicts correctly to the total number of samples. F1 score is a harmonic mean of the recall rate and precision rate of the model.

4.4 Baseline Model

GCN [30]: This paper proposed a graph convolution neural network model. Based on the first order approximation derivation of spectral convolution, the convolution operation in image processing is successfully extended to the graph structure. The model takes the node feature information and the adjacency matrix of the graph as the input, and finally applies the output node representation to the node classification task on the homogeneous network.homogeneous network.

GraphSAGE [31]: The learning result in this article is not the feature vector of each node but rather an "aggregation function". Based on the known features and neighbor relationships of each node, it is easily possible to obtain a representation of a new node.

LGCL [32]: After selecting initial nodes, this article uses dynamic random selection with width-based search to select a dynamic number of external nodes, and stops expanding when the required number is reached. After forming the sub-graph, LGCL constructs new neighbor feature vectors based on the original ranking of neighbor node features. The dimension of the feature vector is equal to the number of channels, and convolution operations are performed through a one-dimensional convolutional neural network to compress the multi-dimensional features of the central node and neighbor nodes into a one-dimensional format.

HAN [33]: This paper uses a hierarchical attention mechanism, which uses node-level attention to learn the weights between nodes and their neighbor nodes based on meta-paths, and then uses semantic-level attention to aggregate different weights between multiple meta-paths.

4.5 Comparison and Analysis of Experimental Results

The model proposed in this paper achieved good experimental results on the dataset, as did the baseline models. As shown in Table 2, the accuracy of the model proposed in this paper improved by 2.18% compared to the baseline models, and the F1 score improved by an average of 1.45% compared to the best benchmark method.

Table 2: Different models accuracy, F1

Metrics	Training size	GCN	GraphSAGE	LGCL	HAN	Ours
Accuracy	20%	0.8465	0.8531	0.8421	0.8293	**0.8659**
	40%	0.8501	0.8569	0.8468	0.8331	**0.8712**
	60%	0.8545	0.8631	0.8512	0.8409	**0.8849**
F1	20%	0.8527	0.8520	0.8342	0.8095	**0.8640**
	40%	0.8533	0.8541	0.8429	0.8120	**0.8701**
	60%	0.8621	0.8658	0.8506	0.8284	**0.8803**

As a deep learning model, GCN has achieved good performance on datasets by leveraging homogeneous graph extraction to enhance the classification of heterogeneous nodes. However, this approach fails to capture comprehensive information. GraphSAGE randomly selects nodes for aggregation, which can lead to the omission of important neighbor nodes, resulting in insufficient information. LGCL selects an initial node and dynamically and randomly picks peripheral nodes through a breadth-first search. It constructs new neighbor feature vectors based on the ranking of original neighbor node features and performs convolutional operations using a one-dimensional convolutional neural network. This compresses the multi-dimensional features aggregated from central and neighbor nodes into a one-dimensional format. LGCL improves network coverage by increasing the number of randomly selected nodes, but its performance relies on the manually set quantity. HAN aggregates nodes using node-level attention based on adjacent nodes through meta-paths. It further aggregates the representations of multiple meta-paths using semantic-level attention. However, HAN is limited by the capability of meta-paths to model higher-order structures and semantic information, which prevents it from obtaining higher-quality representations. In this paper, when aggregating and updating user nodes, we consider the complex relationships among hidden neighbor nodes and higher-order neighbor nodes. We employ an attention mechanism for importance selection. Consequently, our proposed model achieves promising classification results on the dataset.

4.6 Detailed Analysis of the Model

The preference propagation fusion layer consists of three main components: the K-order neighbor node selection module, the hidden neighbor node selection module, and the preference propagation depth. Three sets of ablation experiments were designed to demonstrate the importance of these three components in the model. The results of the ablation experiments are shown in Table 3. In the first set of ablation experiments, the K-order neighbor node selection module was removed from the original model while keeping the other functionalities intact. The K-order neighbor node selection module enhances the higher-order structure of nodes and increases the social depth of users during node updates. The second set of ablation experiments builds upon the first set by removing the implicit neighbor node selection module. This module calculates implicit neighbor nodes based on similarity and aims to incorporate important information during node fusion, thereby improving the social breadth of users. The third set of ablation experiments further removes the preference propagation depth module. An appropriate preference propagation depth N can effectively simulate user social preferences and improve the overall performance of the model. This section's ablation experiments further demonstrate the effectiveness of the model's performance and the formulation of access control policies.

Table 3: Ablation experiments

Method	Accuracy	F1
Ours	0.8849	0.8803
K-order neighbor node selection module	0.8674	0.8530
K-order neighbor node selection module hidden neighbor node selection module	0.8352	0.8299
K-order neighbor node selection module hidden neighbor node selection module preference propagation depth	0.8063	0.8003

5 Conclusion

Existing social network access control schemes often overlook user preferences, making it challenging to recommend optimal access control strategies to users. In this paper, based on the social network scenario, we construct heterogeneous and homogeneous graphs to simulate user personal preferences and user social preferences, respectively. We design a deep learning-based access control model that incorporates user preferences. To address the issue of different feature spaces for nodes of different types in the heterogeneous graph, we employ an attention mechanism to dynamically aggregate nodes of different types. In the homogeneous graph, during node updates, we broaden the scope of node selection by designing a high-order neighbor node module and a hidden neighbor node selection module. These modules consider high-order and hidden neighbor nodes, expanding the breadth and depth of user preferences, resulting in more comprehensive aggregation of node information. Furthermore, we apply multiple evaluation metrics to real-world datasets to demonstrate the effectiveness of the proposed model. The experimental results confirm that the access control strategies formulated by our model exhibit high accuracy. As future work, we plan to further study personalized social access control models. In this paper, we use unimodal data, and the next step will be to integrate multimodal data for more comprehensive and complete analysis.

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 is supported by the National Natural Science Foundation of China Project (No. 62302540); The Open Foundation of Henan Key Laboratory of Cyberspace Situation Awareness (No. HNTS2022020); Natural Science Foundation of Henan Province Project (No. 232300420422); The Natural Science Foundation of Zhongyuan University of Technology (No. K2023QN018); Key Research and Promotion Project of Henan Province in 2021 (No. 212102310480).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Fuyang Li, Fangfang Shan; data collection: Fuyang Li; analysis and interpretation of results: Fuyang Li, Fangfang Shan, Peiyu Ji, Mengyi Wang; draft manuscript preparation: Fuyang Li,

Fangfang Shan, Huifang Sun; Writing-review and editing: Fuyang Li, Fangfang Shan, Zhenyu Wang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

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

References

1. Nuhil Mehdy, A., Mehrpouyan, H. (2020). A user-centric and sentiment aware privacy-disclosure detection framework based on multi-input neural network. *PrivateNLP 2020 Workshop on Privacy and Natural Language Processing Colocated with 13th ACM International WSDM Conference*, pp. 21–26. Houston, USA.
2. Nobi, M. N., Krishnan, R., Huang, Y., Shakarami, M., Sandhu, R. (2022). Toward deep learning based access control. *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy*, pp. 143–154. Baltimore, USA.
3. Xu, L., Jiang, C., He, N., Han, Z., Benslimane, A. (2018). Trust-based collaborative privacy management in online social networks. *IEEE Transactions on Information Forensics and Security, 14(1),* 48–60.
4. Voloch, N., Gal-Oz, N., Gudes, E. (2021). A trust based privacy providing model for online social networks. *Online Social Networks and Media, 24,* 100138.
5. Wang, H., Cao, J., Zhang, Y., Wang, H., Cao, J. et al. (2020). Trust-based access control management in collaborative open social networks. *Access Control Management in Cloud Environments,* 203–221.
6. Aparna, M., Nalini, N. (2020). A novel access control for cloud services using trust based design. *Inventive Computation Technologies, 4,* 702–710.
7. Susan, S. P., Sarath, G. (2019). A trust network driven user authorization scheme for social cloud. *Recent Findings in Intelligent Computing Techniques: Proceedings of the 5th ICACNI 2017*, pp. 533–542. Singapore.
8. Xu, G., Liu, B., Jiao, L., Li, X., Feng, M. et al. (2020). Trust2privacy: A novel fuzzy trust-to-privacy mechanism for mobile social networks. *IEEE Wireless Communications, 27(3),* 72–78.
9. Qiu, M., Gai, K., Thuraisingham, B., Tao, L., Zhao, H. (2018). Proactive user-centric secure data scheme using attribute-based semantic access controls for mobile clouds in financial industry. *Future Generation Computer Systems, 80,* 421–429.
10. Ma, T., Su, Y., Rong, H., Qian, Y., Al-Nabhan, N. (2022). Rule fusion of privacy protection strategies for co-ownership data sharing. *Mathematics, 10(6),* 969.
11. Masoumzadeh, A., Narendran, P., Iyer, P. (2021). Towards a theory for semantics and expressiveness analysis of rule-based access control models. *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*, pp. 33–43. Spain.
12. Dundua, B., Kutsia, T., Marin, M., Rukhaia, M. (2020). Specification and analysis of abac policies in a rule-based framework. *Applications of Mathematics and Informatics in Natural Sciences and Engineering: AMINSE 2019*, pp. 101–116. Tbilisi, Georgia.
13. Ma, L., Yang, W., Huo, Y., Zhong, Y. (2018). Research on access control model of social network based on distributed logic. *Future Generation Computer Systems, 83,* 173–182.
14. Marin, M., Kutsia, T., Dundua, B. (2019). A rule-based approach to the decidability of safety of abacα. *Proceedings of the 24th ACM Symposium on Access Control Models and Technologies*, pp. 173–178. Toronto, ON, Canada.
15. Karimi, L., Aldairi, M., Joshi, J., Abdelhakim, M. (2021). An automatic attribute-based access control policy extraction from access logs. *IEEE Transactions on Dependable and Secure Computing, 19(4),* 2304–2317.

16. Yahiatene, Y., Menacer, D. E., Riahla, M. A., Rachedi, A., Tebibel, T. B. (2019). Towards a distributed abe based approach to protect privacy on online social networks. *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–7. Marrakesh, Morocco.
17. Wei, W., Liu, S., Li, W., Du, D. (2018). Fractal intelligent privacy protection in online social network using attribute-based encryption schemes. *IEEE Transactions on Computational Social Systems, 5(3),* 736–747.
18. Li, J., Chen, X., Chow, S. S., Huang, Q., Wong, D. S. et al. (2018). Multi-authority fine-grained access control with accountability and its application in cloud. *Journal of Network and Computer Applications, 112,* 89–96.
19. Safi, S. M., Movaghar, A., Ghorbani, M. (2022). Privacy protection scheme for mobile social network. *Journal of King Saud University-Computer and Information Sciences, 34(7),* 4062–4074.
20. Hou, Y., Liu, W., Lin, H., Wang, X. (2020). Multi-layer access control mechanism based on blockchain for mobile edge computing. *2020 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pp. 285–291. Exeter, UK.
21. Ma, K., Yang, G., Xiang, Y. (2020). RCBAC: A risk-aware content-based access control model for large-scale text data. *Journal of Network and Computer Applications, 167,* 102733.
22. Li, F., Sun, Z., Niu, B., Cao, J., Li, H. (2019). An extended control framework for privacy-preserving photo sharing across different social networks. *2019 International Conference on Computing, Networking and Communications (ICNC)*, pp. 390–394. Honolulu, USA.
23. Sun, Z., Wan, J., Yin, L., Cao, Z., Luo, T. et al. (2022). A blockchain-based audit approach for encrypted data in federated learning. *Digital Communications and Networks, 8(5),* 614–624.
24. Yin, L., Lin, S., Sun, Z., Li, R., He, Y. et al. (2023). A game-theoretic approach for federated learning: A trade-off among privacy, accuracy and energy. *Digital Communications and Networks.* https://doi.org/10.1016/j.dcan.2022.12.024
25. Akkuzu, G., Aziz, B., Adda, M. (2020). Towards consensus-based group decision making for co-owned data sharing in online social networks. *IEEE Access, 8,* 91311–91325.
26. Sharma, P., Jindal, R., Borah, M. D. (2022). Blockchain-based cloud storage system with CP-ABE-based access control and revocation process. *The Journal of Supercomputing, 78,* 7700–7728.
27. Shan, F., Li, H., Zhu, H. (2018). Game theory based forwarding control method for social network. *Journal on Communications, 39(3),* 172–180.
28. Hu, D., Hu, C., Fan, Y., Wu, X. (2018). oGBAC–A group based access control framework for information sharing in online social networks. *IEEE Transactions on Dependable and Secure Computing, 18(1),* 100–116.
29. Liu, X., Ji, K., Fu, Y., Du, Z., Yang, Z. et al. (2021). P-Tuning v2: Prompt Tuning can be comparable to fine-tuning universally across scales and tasks. arXiv preprint arXiv:2110.07602, 2022.
30. Kipf, T. N., Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.
31. Hamilton, W. L., Ying, R., Leskovec, J. (2017). Inductive representation learning on large graphs. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1025–1035. Red Hook, USA.
32. Gao, H., Wang, Z., Ji, S. (2018). Large-scale learnable graph convolutional networks. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1416–1424. London, UK.
33. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y. et al. (2019). Heterogeneous graph attention network. *The World Wide Web Conference*, pp. 2022–2032. New York, USA.