ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

Neural network driven teaching resource usage behavior modeling is used for university teaching system access control

Qian Xiao[1,*]

[1] Tianjin Univercity Of Commerce Cooperative School of International Education

SUMMARY: *The resource access behavior in university teaching system has time continuity, role difference and context dependence, which affect the discrimination accuracy and implementation stability of access control. This paper proposes a neural network driven approach for modeling access behavior. Based on the semester log of a university teaching platform, a data set containing 1.26 million access records, 42,380 valid sessions, 18 types of teaching resources and 6-level permission labels was constructed. By jointly encoding the user role, request interval, resource sensitivity level, terminal type and historical path, the session-level behavior representation is generated by combining the gated recursive unit, graph relationship propagation and attention aggregation mechanism, and the permission discrimination and control execution are completed. Experimental results show that the accuracy of access level identification reaches 97.3%, the F1 value reaches 96.1%, the average response delay is 21 ms, and the unauthorized misjudgment rate is reduced to 0.18%. This method can maintain stable control performance in different teaching stages and different request sizes, and support fine-grained, real-time and computable control of teaching resources access in university teaching systems.*

KEYWORDS: *Neural network; Teaching resource usage behavior; Access control; College Teaching System*

1 Introduction

The resource invocation in university teaching system has been extended from single signon verification to compound access link covering course space, file warehouse, experimental platform, assignment interface and mobile terminal, and the resource usage behavior shows obvious time continuity, role difference and context dependence. The computational modeling of teaching resources usage behavior can depict the access trajectories of teachers, students and administrators in different task stages, and provide a continuous basis for permission allocation, request verification and control execution. Access control in such systems no longer stays at static authorization, but needs to combine resource sensitivity level, session environment, terminal state and historical operation sequence to complete finer-grained discrimination. Therefore, neural network modeling for behavior flow has become a feasible technical path in the security control of university teaching systems. Compared with the general service platform, the resource access in the university teaching system has the characteristics of tile-driven, semester segmentation, course correlation and task dependence, and the authority boundaries of the same subject are not consistent in the links of course selection, teaching, submission, marking and filing. Only when the continuous behavior

*sissixiao@126.com
https://doi.org/10.65102/is2026212
Published: 30 April, 2026

pattern is incorporated into the control link, the access discrimination results can be consistent with the real teaching process. This linkage mode from behavior expression to permission execution is more in line with the computing requirements of multi-role, multi-resource and multi-session concurrent running in the teaching information system. At the same time, it is convenient for subsequent real-time deployment. Easy to audit. Also traceable

In related research, You et al. studied the online learning access control decision framework supported by knowledge graph, which organized resource entities, user relationships and decision semantics into a unified representation space, providing structural support for permission inference in education scenarios [1]. Shan et al. proposed weighted GraphSAGE context-aware access control method, which deals with context dependence in big data environment through neighborhood aggregation, and provides a graph representation basis for dynamic authorization computation [2]. Sun et al. studied the blockchain access control protocol in mobile edge cloud collaborative resource sharing to make the resource call records in distributed scenarios verifiable and consistent [3]. Roslin Dayana et al. proposed a trust-aware encrypted role-based access control scheme, which combined trust constraints and role authorization mechanism into the cloud storage control process [4]. Turkea et al. studied the dynamic access control decision execution method based on multi-layer hybrid deep learning in the BYOD environment, which enhanced the authorization adaptation ability under heterogeneous terminal access conditions [5]. Shan F et al. proposed a deep learning social network access control model based on user preferences, which maps behavior tendencies into permission selection basis and extends the idea of individual behavior modeling in access control [6]. Alazab et al. studied the intrusion detection method combined with dynamic access control algorithm, which made node-side access judgment and security response form a linkage calculation mechanism [7]. Liu et al. proposed a secure data sharing method for federated learning based on blockchain aggregation, which provides a new realization path for control trustworthiness in multi-node collaborative environment [8]. Pritee et al. systematically summarized the application ways of machine learning and deep learning in identity authentication and authorization, and clarified the computational connections between behavioral characteristics, authentication logic, and control strategies [9]. Wang et al. studied the behavior authentication method for security scenarios and showed that continuous behavior characteristics can be used as an important basis for access subject identification and control verification [10]. Budžys et al. proposed a deep learning authentication method for insider threat detection of critical infrastructure, which further showed that the sequence behavior representation had stable discrimination value in a high-level security environment [11].

Based on the above research progress, based on the access log of university teaching system, this paper introduces a neural network driven behavior sequence encoding method to jointly represent the time interval, resource category, user identity, operation path and session context in the access request. On this basis, an access discrimination and control execution mechanism for permission constraints is established. The research content of this paper includes the behavior feature coding of the teaching resource usage sequence, and the permission discrimination and control execution analysis for the access control of the teaching system in colleges and universities.

2 Related Research

Access control of teaching resources relies on the stable description of user behavior, and continuous authentication, behavior biometrics recognition and anomaly access detection

have provided a transferable technical basis for this kind of task. Sağbaş et al. studied a continuous authentication system based on soft keyboard keystroke behavior and motion sensor data, and proposed a behavior authentication process driven by machine learning, which extended the mobile terminal identity recognition from one-time login verification to continuous discrimination within a session [12]. Wyciślik et al. proposed a keystroke dynamics biometric method based on deep learning, which strengthened the feature extraction ability of timing tapping patterns on open data sets, and said that the detailed granularity input rhythm could support stable identity discrimination [13]. Vegas et al. studied a deep learning user recognition system based on door handle sensors, which integrated the touch action, pressure response and sensing signals into a unified recognition framework, and expanded the modeling way of behavior characteristics in non-keyboard environments [14]. This kind of research shows that the micro-action sequence of users in the process of device interaction has the properties of computable and traceable, and also shows that the rhythm characteristics of the access subject in continuous operation can be transformed into stable identity expression. The click path, resource residence time, access interval and terminal switching in the university teaching platform also belong to continuous behavior signals. Therefore, the sequence modeling idea formed in the field of behavior authentication can be transferred to the teaching resources access control scenario. In order to more clearly present the characteristics of the research in this direction, the related results can be summarized in Table 1.

Table 1: Summary of research on continuous authentication and behavior recognition

Author	Research Content	Referable Point
Sağbaş et al.	Continuous authentication and soft-keyboard behavior recognition	Continuous in-session verification
Wyciślik et al.	Deep learning-based keystroke dynamics recognition	Fine-grained temporal feature extraction
Vegas et al.	Multi-sensor fusion for user identification	Joint representation of heterogeneous signals

As can be seen from Table 1, although the existing studies originate from different application environments, they all regard the continuous behavior stream as an important carrier for identity identification. This kind of method no longer only relies on static account attributes, but completes the subject characterization through local fluctuations, rhythm differences and operation continuity in time series. This idea has direct reference value for the teaching system in colleges and universities, because the access to teaching resources is not an isolated single event, but a complete link composed of course entry, information browsing, file download, assignment submission and permission jump. As long as the link can be encoded effectively, access control can be extended from static authorization to dynamic discrimination under behavior constraints.

In the access security scenario, the research focus further turns to subject identification, abnormal access detection and graph structure reasoning. Tao et al. proposed an internal user authentication method based on improved temporal convolutional network, which uses mouse dynamic sequence to extract user features and combines one-class support vector machine to complete identity verification [15]. Bin Sarhan et al. studied the insider threat detection method based on machine learning, and showed that the combination of access offset, operation density and context in behavior logs can form the basis for threat discrimination [16]. Roy et al. proposed GraphCH deep framework to map network behavior and human activity attributes into graph structure, which was used to identify abnormal subjects and

insider threats [17]. Gong et al. systematically reviewed the research on schema insider threat detection and pointed out that graph representation is more suitable for describing multi-entity, multi-relationship and stage interaction links in an organization [18]. Xiao et al. proposed a robust anomaly detection method based on graph neural network to enhance the anomaly recognition ability in complex audit data through the multilateral weight relationship graph [19]. Villarreal-Vasquez et al. proposed an anomaly detection framework based on LSTM, which maintained strong prediction and recognition ability in sequential events [20]. Together, these studies show that access control should not be decided only by static role or single request, and the behavior sequence, relationship structure and context evolution should be taken into the decision-making process, so that the control result is closer to the real operation state. In order to further compare the differences in modeling objects and implementation paths of this direction method, related studies can be organized as Table 2.

Table 2: Summary of anomaly access detection and graph modeling research

Author	Research Content	Methodological Focus	Applicability Insight
Tao et al.	Insider user authentication	Temporal convolution and sequential features	Access trajectory encoding
Bin Sarhan et al.	Insider threat detection	Machine learning-based log classification	Multi-dimensional behavioral combination analysis
Roy et al.	GraphCH framework	Joint graph construction of network behavior and human factors	Subject relationship representation
Gong et al.	Review of graph-based insider threat detection	Multi-entity correlation analysis	Graph-structured control modeling
Xiao et al.	Graph neural network-based anomaly detection	Multi-edge-weight relational graph	Complex log discrimination
Villarreal-Vasquez et al.	LSTM-based anomaly detection	High-dimensional temporal modeling	Continuous access prediction

Combined with Table 2, it can be found that the existing research has extended from single identity recognition to relational network reasoning and abnormal access recognition, and the modeling object extends from individual users to multiple connections between users, devices, tasks and resources. This change shows that it is difficult to fully cover the control requirements in high-concurrency, multi-node and heterogeneous resource environments by only relying on role rules for access judgment. For the university teaching system, the access behavior is not only related to the subject identity, but also closely related to the course stage, resource sensitivity level, terminal type, time window and historical path. The resource visibility range of teachers, students and managers in different teaching tasks is not consistent with the executable action, and the same subject will show different access patterns in different stages of the semester. If these conditions are handled separately, it is difficult to maintain consistency between control execution and real service links. Therefore, based on the existing results of continuous authentication, anomaly detection and graph structure reasoning, the research on access control for university teaching systems needs to introduce a neural network driven behavior sequence encoding method, which can map the time, type, path and context constraints in resource access links into a learnable representation, and

realize access discrimination and control execution oriented to permission constraints. This method not only preserves the enforceability of access control, but also enhances the consistency between fine-grained authorization calculation and teaching business process, which provides a method basis for subsequent modeling design.

3 Neural network driven access control modeling design of teaching resources usage behavior

3.1 Coding sequence behavior characteristics of teaching resources use based on neural network

The sequence behavior feature coding of teaching resources is used to transform the discrete access records in the university teaching system into a computable continuous representation. The resource call in the platform is not an isolated event, but an ordered link composed of course entry, page stay, information view, file download, assignment submission, score query and permission jump. Different subjects will form different access trajectories in different course stages, and these trajectories carry the information of role attributes, resource types and time intervals. In order to make the subsequent access control complete the finer discrimination according to the behavior change, this paper serializes the access log first, and then uses the neural network to complete the multi-dimensional behavior feature coding. The encoded result not only preserves the semantic meaning of the request itself, but also characterizes the access pattern.

In the stage of raw data organization, it is necessary to define the set of access events of a single user in a session. Let the sequential access sequence of user u in session s be given by equation (1).

$$X_{u,s} = \{x_1, x_2, \dots, x_T\} \tag{1}$$

Here, $X_{u,s}$ represents the access sequence formed by user u within session s, x_t represents the t access event, and T represents the total number of events. This formula gives the basic structure of the neural network input, so that the subsequent encoding can be carried out around the complete access link.

After completing the event definition, vectors need to be constructed for each access action. The log fields in the university teaching system usually include role identity, resource identification, operation type, terminal source, course number and page location. These fields have different dimensions and cannot be directly entered into the same calculation level. In this paper, embedding mapping and linear projection are combined to compress the multi-source fields into the same feature space, as shown in Equation (2).

$$e_t = [r_t \| q_t \| a_t \| d_t \| p_t] W_e + b_e \tag{2}$$

where,e_t represents initial feature vector, r_t represents user role embedding, q_t represents resource category and sensitivity level embedding, a_t represents operation type embedding, d_t represents terminal type embedding, p_t represents session location embedding, W_e and b_e represent mapping parameters. This formula compreszes the heterogeneous log fields into the same feature space, which is convenient for subsequent sequential learning.

The access activity of the teaching system has a temporal structure. The centralized login before class, the information switch in class, the assignment submission after class and the query in the examination stage all reflect different time interval characteristics. In order to

preserve the attenuation relationship and periodic fluctuation at the same time, this paper constructs the time mapping vector, as shown in equation (3).

$$\tau_t = [\exp(-\Delta_t/\lambda), \sin(2\pi\Delta_t/\rho), \cos(2\pi\Delta_t/\rho)] \quad (3)$$

Here, τ_t represents the time interval encoding vector, Δ_t represents the time difference between the current event and the previous event, λ represents the decay scale, and ρ represents the period scale. This formula not only retains the strength of the access interval, but also retains the time interval rhythm in the teaching scene.

As shown in Fig. 1, the sequence encoding process consists of five steps: log cleaning, event segmentation, field embedding, time alignment, and behavior representation generation. The log cleaning phase removes duplicate requests and invalid jumps. The event segmentation phase reorganizes access links according to users, sessions and time Windows. In the field embedding phase, roles, resources and operations are mapped. The time alignment phase incorporates time interval information. The output goes into the subsequent neural computation.

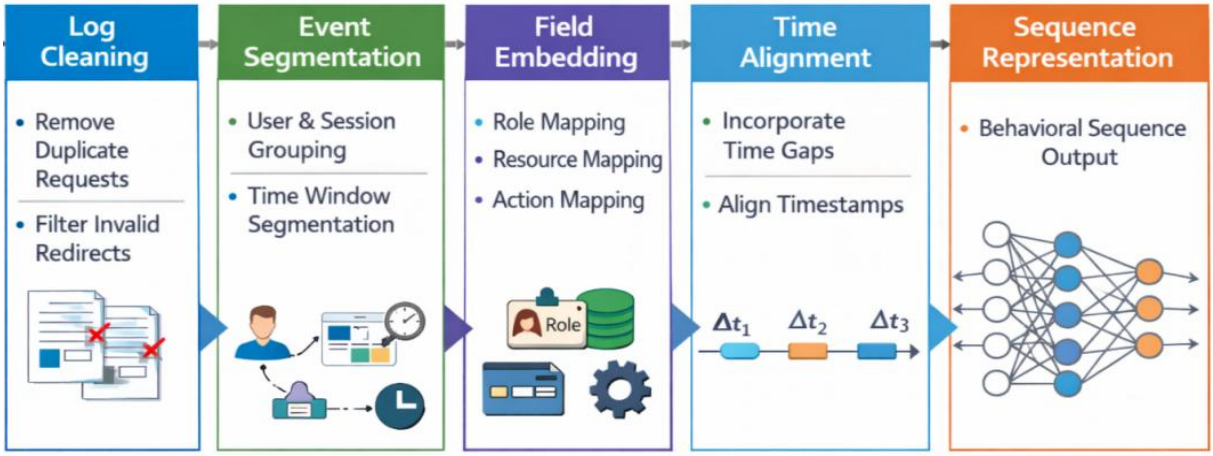

Figure 1: Coding flow of instructional resource usage sequence

After completing the initial vector construction, the gated recurrent unit is used in this paper to extract local sequential dependencies. There are not only short-term continuous browsing, but also cross-module jump in the access sequence of teaching resources. If we only use static aggregation, it is easy to weaken the role of key actions. To this end, in this paper, the current event vector is input into the gating unit together with the time encoding, and its update form is shown in Equation (4).

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tanh(W_h[e_t \| \tau_t] + U_h(r_t \odot h_{t-1})) \quad (4)$$

Here, h_t represents the hidden state at time t, h_{t-1} represents the state at the previous time, z_t represents the update gate, W_h and U_h represent the weight matrix, and $\odot$ represents the element-wise multiplication. This formula is used to balance the current event and historical memory, so that continuous access and critical jumps can be preserved at the same time.

Relying on sequential features alone is still not sufficient to describe structural associations between resources. There are natural links among the course home page, notice page, information directory, assignment entry and score module in the university teaching

system, and the same user's jump path between different resources will also form a repeated pattern. To complement this structural information, this paper constructs an access graph based on resource co-occurrence and page jumps, and uses attention propagation to calculate the neighborhood representation, as shown in Equation (5).

$$g_t = \sigma\left(\sum_{j\in\mathcal{N}(t)} \alpha_{tj} W_g h_j\right), \quad \alpha_{tj} = \frac{\exp(\phi(h_t, h_j))}{\sum_{k\in\mathcal{N}(t)} \exp(\phi(h_t, h_k))} \tag{5}$$

Here, g_t represents the structure representation after graph propagation, $\mathcal{N}(t)$ represents the neighborhood set of the current event, α_{tj} represents the attention weight of neighbor j, W_g represents the graph mapping parameter, and ϕ represents the relevance scoring function. This formula takes resource association and path similarity into the encoding process, so that the representation result has both sequence information and structure information.

After the sequential state and the graph structure state are obtained, it is still necessary to decide which access actions are more representative in the whole session. Permission discrimination in teaching systems usually does not rely on the average results of all events, but more on the access fragments of key resources. Therefore, this paper introduces the attention aggregation mechanism on the fusion state to calculate the session-level summary vector, as shown in Formula (6).

$$\beta_t = \frac{\exp(v^\top \tanh(W_b[h_t \| g_t]))}{\sum_{i=1}^{T} \exp(v^\top \tanh(W_b[h_i \| g_i]))}, \quad c_{u,s} = \sum_{t=1}^{T} \beta_t [h_t \| g_t] \tag{6}$$

Here, β_t represents the aggregated weight of the t event in the whole session, v and W_b represent the attention parameters, and $c_{u,s}$ represent the summary representation of user u in session s. This formula can highlight key jumps and highly sensitive resource access segments, and avoid information dilution caused by average aggregation.

As shown in Fig. 2, the multi-view behavior representation is jointly composed of four parts: sequential state, temporal intensity, resource relationship, and session summary. The sequential state describes the dependence of access actions, the time intensity reflects the request rhythm, the resource relationship represents the linkage path between the page and the document, and the session summary gives the global expression of the whole access. The four representations are fused to form a unified behavior code, which is used as the input of the permission discrimination layer.

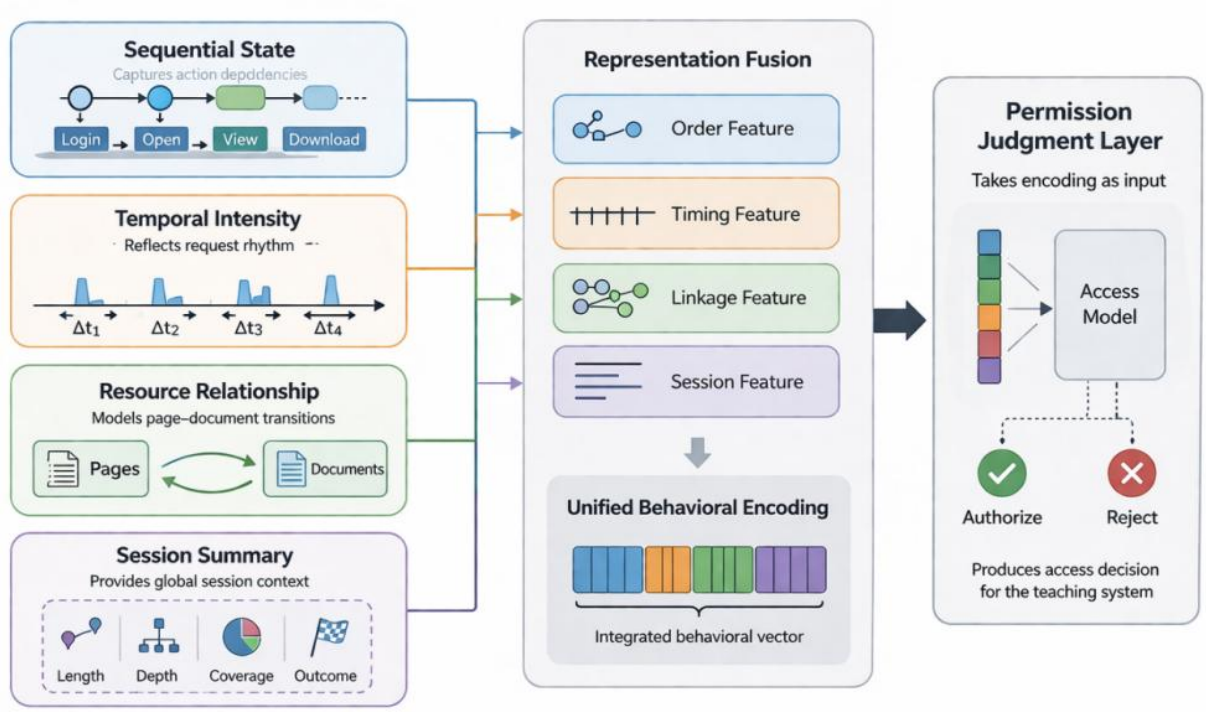

Figure 2: Multi-view instructional resource usage behavior representation structure

In order to enhance the comparability of coding results between different user groups and different course stages, this paper continues to perform normalized projection on the session summary to form the final behavioral feature coding, as shown in Equation (7).

$$z_{u,s} = \mathrm{LayerNorm}(W_z c_{u,s} + b_z + \gamma m_{u,s}) \tag{7}$$

where $z_{u,s}$ represents the behavior coding of the final output, W_z and b_z represent the projection parameters, $m_{u,s}$ represents the statistical vector composed of access frequency, proportion of sensitive resources and abnormal jump rate, γ represents the adjustment coefficient, and LayerNorm represents the layer normalization operation. The expression combines the deep sequence expression and the statistical behavior characteristics into the same output space, so that the encoding result retains not only the representation ability of the neural network, but also the interpretable information in the access log of the teaching system.

After the above processing, the teaching resource usage sequence is transformed into a high-dimensional representation with time continuity, resource structure and behavior difference. The representation can distinguish different access modes such as normal browsing, centralized submission, batch correction and management switching, and provide a stable input basis for the discrimination of permission constraints and the execution of control in the next section.

3.2 Access discrimination and control execution oriented to permission constraints

Permission constraint-oriented access discrimination and control enforcement is used to translate the behavior encoding obtained in the previous section into executable access decisions. Permission judgment in university teaching system is not a dynamic process, which is composed of subject identification, constraint matching, risk calculation, action generation and result writeback. The access requests of users in the course space, assignment module,

data warehouse and grade management interface have different scenarios, and the allowed actions of the same identity under different time periods and resource levels are also different. Therefore, access control needs to combine behavior representation, resource sensitivity and session context on the basis of role rules to form dynamic discrimination results. The behavior code formed in the previous section has been able to describe the time continuity and structural correlation of the access sequence. On this basis, this section constructs a discrimination mechanism of permission constraints, and maps the discrimination results to control execution instructions, so that the teaching system can complete fine-grained access control.

When a single request arrives at the discrimination layer, it is necessary to jointly represent the behavior encoding and the resource constraint vector. Suppose that the user's final behavior in the session is encoded as the output vector of the previous section, and the constraint description of the current access resource is composed of the resource sensitivity level, the course scope, the allowed time window and the terminal limit, then the joint input can be defined as shown in Equation (8).

$$u_t = \left[z_{u,s} \,\|\, k_t \,\|\, o_t \,\|\, \eta_t\right] W_u + b_u \tag{8}$$

Here, u_t represents the joint discriminant input of the t request, $z_{u,s}$ represents the behavior encoding generated in the previous section, k_t represents the resource constraint vector, o_t represents the current operation type vector, η_t represents the terminal and period context vector, W_u and b_u represent the mapping parameters. This formula pushes the behavior state and resource boundary into the unified space at the same time, which provides the basis for the subsequent permission response calculation.

The joint representation not only preserves the continuous behavior characteristics of the access subject in the current session, but also introduces the control boundary of the resource itself, so that the authorization is calculated according to the real-time matching relationship between the subject and the resource instead of relying on a single role label. In this way, the marking behavior of the teacher, the submission behavior of the student, and the configuration behavior of the administrator can be expressed in the same discriminative space, only the corresponding constraint positions are different.

In order to form a stable mapping between the joint input and the permission state, the gated discriminative network is used to calculate the requested permission response value. The network consists of two layers of nonlinear mappings and a gated calibration unit, which is able to compress invalid fluctuations and enforce key limiting terms. Its core calculation is shown in Equation (9).

$$m_t = \sigma(W_g u_t + b_g), \quad p_t = g_t \odot \tanh(W_p u_t + b_p) \tag{9}$$

Here, m_t represents the gating coefficient vector, p_t represents the permission response vector, W_g, W_p, b_g, and b_p represent the discriminant network parameters, $\sigma(\cdot)$ represents the Sigmoid function, and $\odot$ represents element-wise multiplication. The gated branch is used to measure the consistency between the current behavior vector and the constraint vector, and the main branch is used to output the permission score. After multiplying the two branches, the key restriction items can be retained and the occasional noise can be suppressed.

The gated branch in the formula is used to measure the degree of agreement between the current behavior vector and the constraint vector, and the master branch is used to output the permission score. The result obtained after the multiplication of the two branches not only

reflects the acceptability of the access request, but also suppresses the misjudgment caused by a single high-frequency action. The design is suitable for the scene of multi-step continuous access in the teaching system, because a commit is usually accompanied by browse, modify, look back and enter again. The control execution needs to recognize this compound process, instead of treating each click as an independent event.

As shown in Fig. 3, the permission discrimination process consists of five links: request access, behavior reading, constraint matching, discrimination output and action distribution. The request access layer is responsible for receiving page access, interface call and resource download requests. The behavior reading layer fetches the behavior encoding result of the current session. Constraint matching layer loading target resource control conditions; The discriminative output layer calculates three types of results: allowed, restricted and blocked. The action dispatcher converts the result into a pass, double check, or reject instruction. The whole process keeps online execution characteristics, and can complete control decisions before resource access.

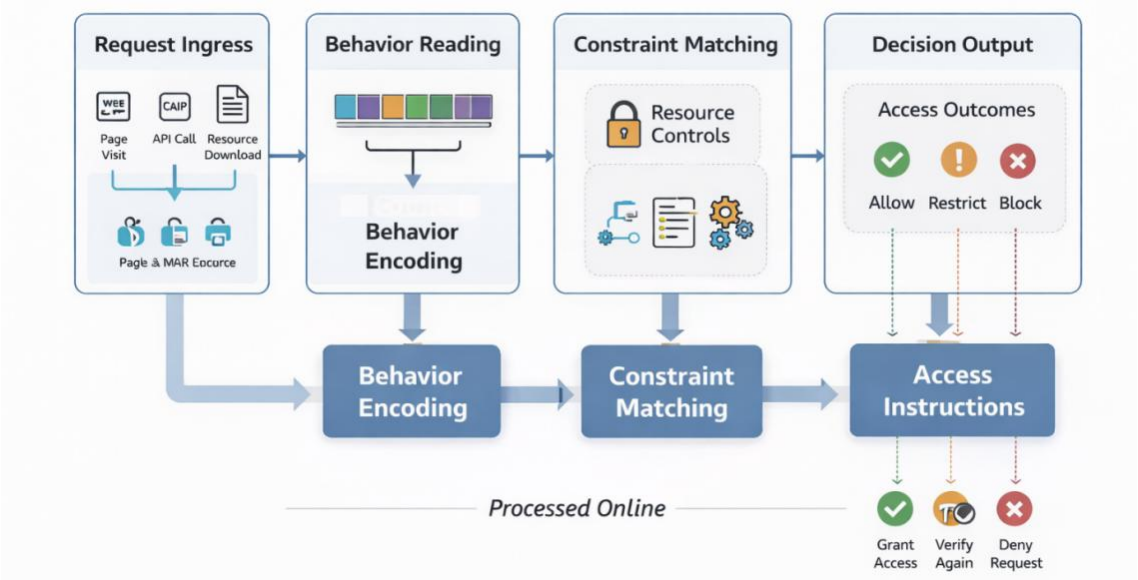

Figure 3: Access discrimination logic for permission constraints

After obtaining the permission response value, it is also necessary to estimate the access risk combined with the abnormal offset information. The abnormal access in the teaching system of colleges and universities is not only manifested as high-frequency requests, but also may be manifested as cross-course jump, short-term batch reading, step-over view and atypical period of time operation. In order to incorporate these factors into the control link, this paper defines the multi-factor risk score function, as shown in equation (10).

$$r_t = \sigma(\alpha \| p_t - \mu_c \|_2^2 + \beta \xi_t s_t + \chi \delta_t) \tag{10}$$

where r_t represents the risk score of the current request, μ_c represents the center of similar normal access pattern, ξ_t represents the resource sensitivity level, s_t represents the role permission level, δ_t represents the offset between time period and terminal, α, β and χ represent the weighting coefficient. This equation jointly evaluates the risk intensity of the current request by behavior offset, resource tension and context offset.

The risk score consists of three parts: one part comes from the offset distance between the behavior encoding and the normal mode center, one part comes from the tension between the resource sensitivity level and the current role permission, and one part comes from the context

shift caused by the time window and the terminal state. After normalization, the risk score is limited between zero and one to facilitate subsequent threshold comparison. The design enables the control layer to pay attention to behavior anomalies, resource constraints and environmental conditions at the same time, which is more suitable for the complex and concurrent access situation in the university teaching platform.

In order to ensure that the discrimination result has a clear hierarchy, this paper does not directly output the binary allowed result, but divides the access status into three execution levels: pass, restricted and block. The execution level is determined in the way shown in Equation (11).

$$a_t = \begin{cases} \text{pass}, & r_t < \theta_1 \wedge \max(p_t) \geq \tau \\ \text{limit}, & \theta_1 \leq r_t < \theta_2 \\ \text{block}, & r_t \geq \theta_2 \end{cases} \tag{11}$$

Here, a_t represents the execution action level, θ_1 and θ_2 represent the risk threshold, and τ represents the permission response threshold. The pass state corresponds to normal resource access, the restricted state corresponds to requests that require additional validation or range narrowing, and the blocked state corresponds to requests that are directly rejected and entered into the audit link.

Fig. 4 illustrates the linkage process of control execution and feedback update. After the discriminative layer outputs the access level, the execution module generates action instructions and writes them to the control cache. If the request is allowed, the result enters the normal resource link; If the request is restricted, additional verification is triggered and the state is recomputed according to the verification result. If the request is blocked, the audit information, alarm tag and session summary are synchronously recorded. The feedback module writes the result back to the user state vector, which can be updated for subsequent requests. In this way, access control is no longer a static rule response, but a closed-loop mechanism with online update capability.

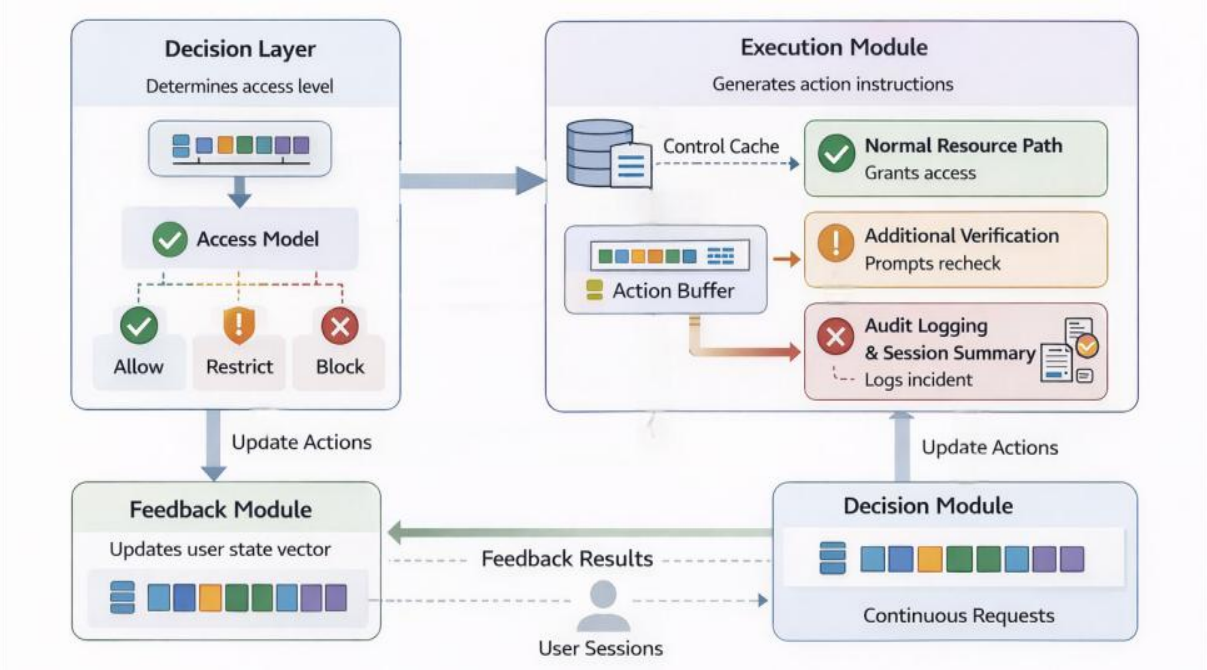

Figure 4: Access control enforcement and feedback update process

The three-level output can handle access requests in teaching scenarios more finely. Through the state corresponding to normal resource access, the system directly released; In the restricted state, the system triggers a second acknowledgment, CAPTCHA verification, or range reduction for requests that are slightly offset but can be completed by additional verification. The blocked state corresponds to high-risk requests that are denied access and written to the audit queue. Such a control method can avoid the interference of one-size fits all refusal to teaching activities, and also make the execution of permissions more in line with the actual business rhythm.

In order to keep the control parameters stable under different courses, different user sizes, and different time loads, this paper introduces a batch optimization objective to jointly constrain the discriminant output and execution deviation. The training loss consists of a classification loss, a risk calibration loss, and an execution consistency loss, as shown in Equation (12).

$$\mathcal{L} = \lambda_1 \mathcal{L}_{ce} + \lambda_2 \frac{1}{B} \sum_{i=1}^{B} (r_i - \hat{r}_i)^2 + \lambda_3 \frac{1}{B-1} \sum_{i=2}^{B} \| p_i - p_{i-1} \|_2^2 \tag{12}$$

Here, $\mathcal{L}$ represents the total loss, $\mathcal{L}_{ce}$ represents the access level classification loss, B represents the batch size, r_i represents the risk score of the model output, $\hat{r}_i$ represents the target risk value corresponding to the audit label, and λ_1, λ_2, and λ_3 represent the loss weight. The third term is used to constrain the output fluctuations between successive requests so that the control results remain stable between adjacent operations.

The classification loss is used to constrain the prediction results of the permission level, the risk calibration loss is used to narrow the difference between the risk score and the real audit label, and the execution consistency loss is used to ensure the smooth change of the control results of adjacent requests in continuous scenarios. After the combined effect of the three types of losses, the model can still maintain high discriminant consistency under the condition of high concurrent access and cross-semester traffic variation, which has direct significance for the actual deployment of the teaching platform.

In the parameter update phase, this paper adopts the adaptive optimization strategy with warm-up term, and uses the execution feedback to correct the key parameters. The parameters are updated as shown in Equation (13).

$$\theta^{(n+1)} = \theta^{(n)} - \eta_n \frac{\hat{m}_n}{\sqrt{\hat{v}_n} + \varepsilon} + \omega \nabla_\theta \mathcal{L}_{fb} \tag{13}$$

where $\theta^{(n+1)}$ represents the model parameters of the n and $n+1$ iterations, η_n represents the current learning rate, $\hat{m}_n$ and $\hat{v}_n$ represent the first and second moment estimates, ε represents the stability term, ω represents the feedback correction weight, $\mathcal{L}_{fb}$ represents the feedback loss constructed based on the real execution results. This formula can re-inject the real release, restriction and blocking results into the parameter update process while maintaining the convergence speed.

This update method maintains a fast convergence speed in the early training stage, and gradually reduces the fluctuation in the subsequent stage, so that the risk estimation and permission discrimination maintain synchronous convergence. The introduction of the execution feedback term enables the model to use the real release, restriction and blocking results to recalibrate the parameters, so as to improve the fit degree of the control layer to the actual access link. After the above processing, the access discrimination and control execution

oriented to permission constraints form a complete computing link from behavior input, constraint mapping, risk estimation, state division to result writeback, which provides a method basis for control effect evaluation in subsequent experimental analysis.

4 Experimental analysis of access control modeling of teaching resource usage behavior driven by neural network

4.1 Analysis of coding results of teaching resource use behavior

In the experimental analysis of teaching resource usage behavior coding, this paper focuses on four aspects of representation quality, category separation, cross-stage stability and resource constraint sensitivity. The experimental data comes from the real operation log of a unified teaching platform in a university in one semester, which contains 1263840 access records, 42380 valid sessions, 18 types of teaching resources and 6 level permission tags, involving typical modules such as course home pages, lecture attachments, experimental documents, assignment entry, score query, discussion boards and management configuration. In order to ensure that the evaluation results can reflect the applicability of behavior coding in real access control links, the experiment splits the samples according to user sessions, and keeps the distribution of teachers, students and administrators in the training set, validation set and test set consistent. In the training stage, the batch input method was used, and the inter-class distance, intra-class compactness and high-sensitive resource aggregation of the embedding space were continuously monitored during the iteration process. When the contour coefficient of the behavior representation on the validation set does not rise for three consecutive rounds, the training is stopped to ensure that the coding results have both separability and stability.

For comparison, the proposed method, BiGRU coding method and Temporal-CNN coding method are selected as the control group, and tested under the same data segmentation conditions. Firstly, the overall representation effects of the three types of methods are statistically analyzed, and the results are shown in Table 3. Four indices are given in the table: cluster purity, contour coefficient, Davies-Bouldin index and normalized mutual information. The contour coefficient is used to describe the balance between intra-class compactness and inter-class separation. The lower Davies-Bouldin index means the clearer boundaries between different classes.

Table 3: Comparison of overall representation results for different behavior encoding methods

Method	Clustering Purity	Silhouette Coefficient	Davies–Bouldin Index	Normalized Mutual Information
Proposed Method	0.913	0.472	0.684	0.901
BiGRU	0.864	0.391	0.918	0.842
Temporal-CNN	0.836	0.358	1.047	0.817

From Table 3, we can see that the proposed method shows better comprehensive results on all four indicators. The cluster purity reaches 0.913, indicating that similar access behavior has a higher concentration in the embedding space. The silhouette coefficient reaches 0.472, which indicates that the intra-class distance is smaller and the inter-class distance is larger. The Davies-Bouldin index decreases to 0.684, indicating a clearer boundary between different access patterns. The normalized mutual information reaches 0.901, which indicates that there is a strong agreement between the behavior encoding and the true label. The results show that

the multi-view sequence encoding method constructed in this paper can simultaneously maintain time continuity, resource correlation and role difference, thereby enhancing the support ability of action representation for subsequent access control.

As shown in Fig. 5, this paper further employs a two-dimensional embedded distribution map to visualize four typical access patterns, including normal browsing, centralized submission, batch correction, and high-privilege switching. In the figure, each dot represents a session-level behavior vector, and different colors represent different access patterns. From the perspective of spatial distribution, the boundaries of the four clusters formed by the proposed method are relatively clear, especially for the two complex behaviors of high-authority switching and batch grading, the overlap area between embedded points is significantly reduced. In contrast, the BiGRU method has a large overlap between centralized submission and normal browsing, and the Temporal-CNN method shows a more obvious stretch distribution in the high-authority switching region. The visualization results corroborate each other with the silhouette coefficient and cluster purity results in Table 3, indicating that the proposed method has stronger structural identification ability when dealing with multi-step continuous access links.

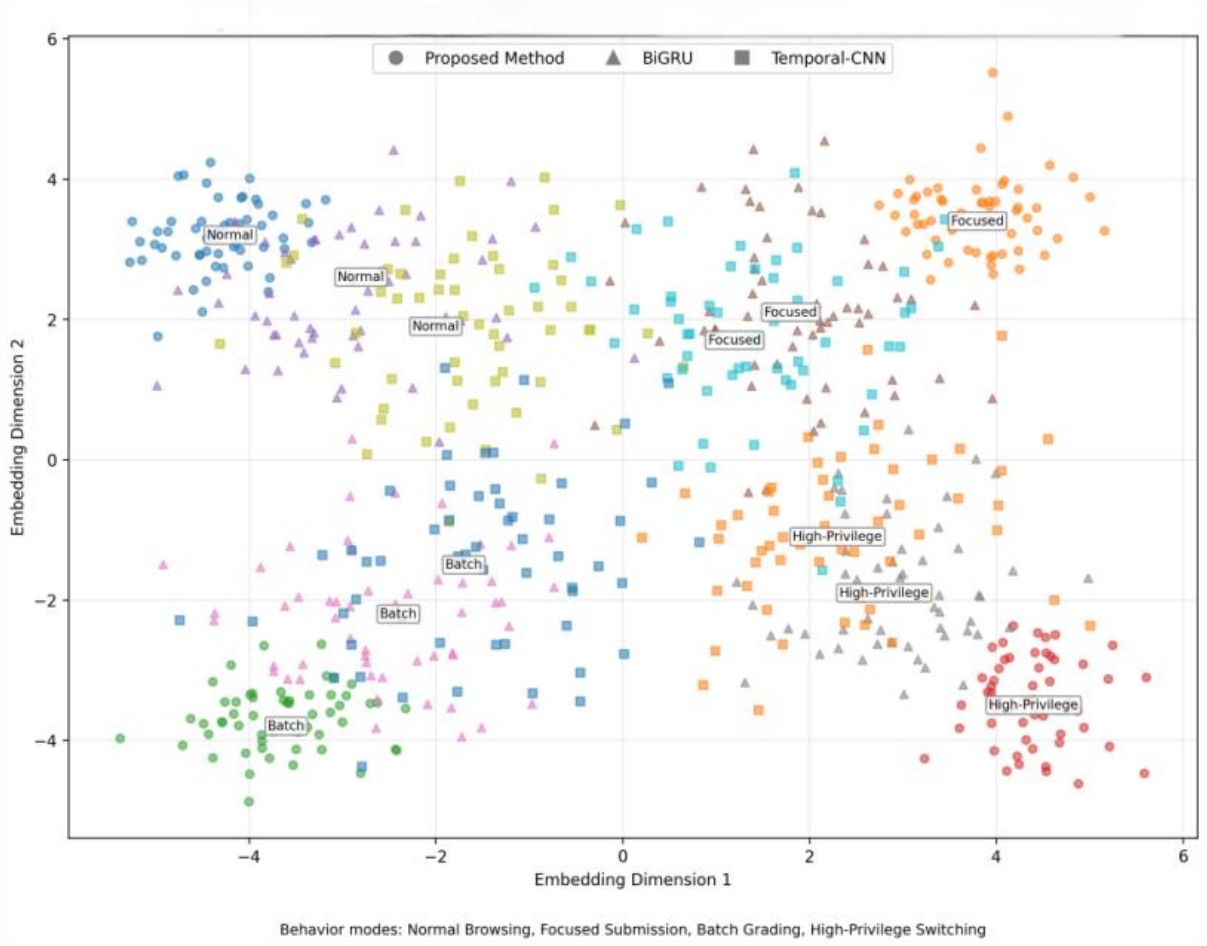

Figure 5: Comparison of embedding distributions for typical access patterns under different encoding methods

After the overall representation capability validation, the experiment continues to analyze how sensitive the behavior encoding is to resource constraints. Because the access control in the teaching system is highly related to the resource type, access period and permission level,

if the coding results cannot reflect the linkage relationship of these factors, the control effect of the subsequent discrimination layer will be limited. To this end, this paper counts the average attention weights of different resource categories in different time periods, and draws the period-resource attention heat map, as shown in Fig. 6. The results show that the attention intensity of the proposed method is more concentrated on the three types of highly constrained resources, such as assignment entrance, score query and management configuration, and a clearer hotspot area is formed in the assignment concentration period and evening access period. The attention distribution of general information browsing and course homepage is relatively smooth, indicating that the model can automatically adjust the focus of feature aggregation according to the resource sensitivity level and access stage. This phenomenon indicates that behavior coding does not simply record access frequency, but effectively retains the traction effect of resource constraints on access paths in the embedding process.

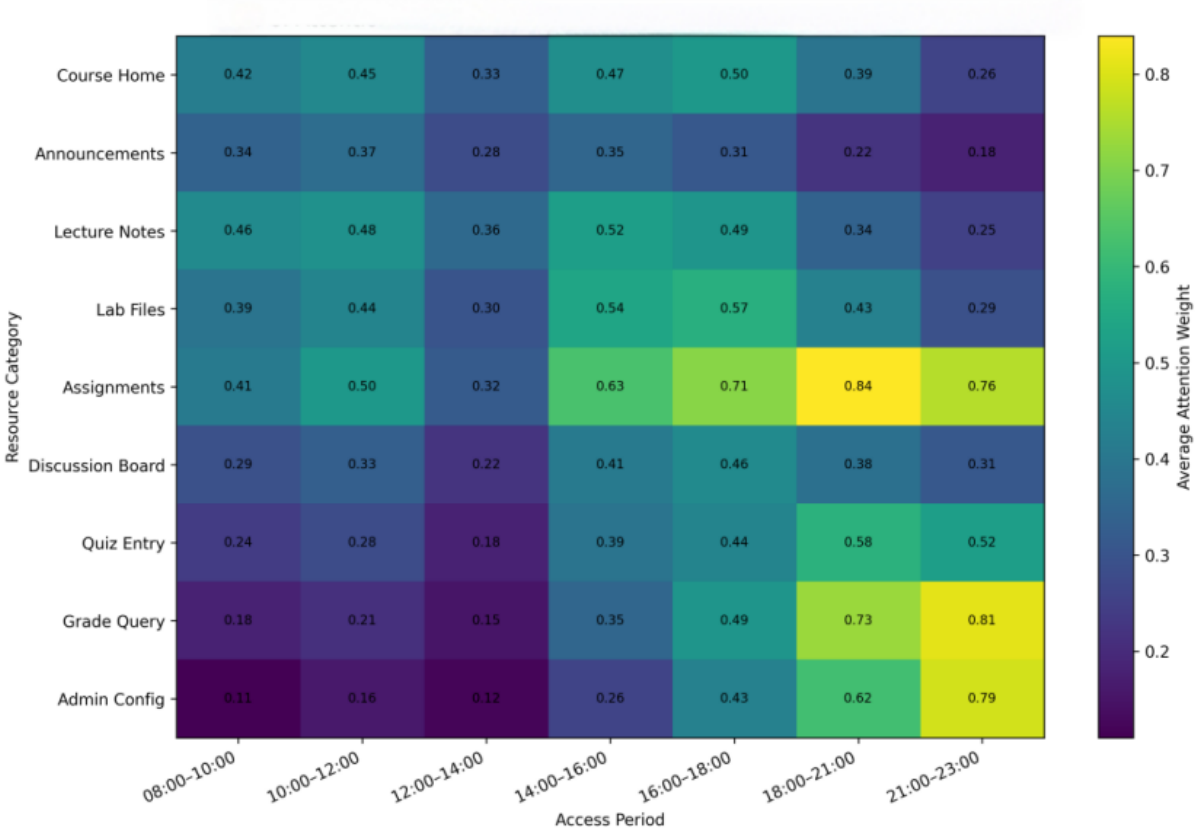

Figure 6: Resource category -attention weight thermal distribution map during access period

In order to further observe the stability of coding results in different teaching stages, the experiment divided the test sample into four stages: the beginning of the course, the usual teaching period, the assignment concentration period and the final assessment period, and counted the representation variance, resource-sensitive correlation coefficient and cross-stage drift distance of each stage. The results are shown in Table 4. Representation variance is used to measure the dispersion degree of coding results in the same stage, resource sensitive correlation coefficient reflects the connection strength between behavior vector and resource sensitive level, and cross-stage drift distance describes the change range of embedding center in adjacent stages.

Table 4: Behavior coding stability results of the proposed method in different teaching stages

Teaching Stage	Representation Variance	Resource Sensitivity Correlation Coefficient	Cross-Stage Drift Distance
Initial Course Stage	0.021	0.742	0.000
Regular Teaching Stage	0.024	0.758	0.083
Assignment-Intensive Stage	0.029	0.776	0.117
Final Assessment Stage	0.031	0.781	0.126

Table 4 shows that the representation variance of the proposed method on the four stages always remains at a low level, with the highest being only 0.031, indicating that the coding results still have strong stability in different teaching stages. The resource-sensitive correlation coefficient gradually increased from 0.742 to 0.781, indicating that with the increase of the complexity of teaching tasks, the behavior representation had a more obvious response to the constraint of high-sensitive resources. Although the cross-stage drift distance increases in the job concentration period and the final assessment period, the overall amplitude is still in the controllable range, indicating that the model can adapt to the phasic changes of access density and operation mode without the drastic drift of embedding space.

The experimental results of teaching resource usage behavior coding show that the method in this paper has formed a strong consistent expression in the three dimensions of access pattern separation, resource constraint response and phase stability. The boundaries of normal browsing, centralized submission, batch correction and high-privilege switching in the embedded space are clearer, which indicates that the joint modeling of sequence state, time intensity and resource relationship can effectively describe the access differences in teaching systems. The high sensitive resources show higher correlation coefficient and more concentrated attention distribution in the job concentration period and the final assessment period, indicating that the coding results are no longer limited to the statistics of access frequency, but can reflect the constraint effect of resource level on behavior path. The representation variance remained in a low range under different teaching stages, which also showed that the coding layer still had good stability under the conditions of access density changes and operation rhythm switching. Therefore, the behavior vector obtained in this section can be used as the direct input of the subsequent permission discrimination layer, so that the access control process can complete the joint identification of the subject state, the resource boundary and the operation context before the request arrives.

4.2 Permission discrimination and access control effect analysis

In the analysis of permission discrimination and access control effect, the experiment continues to use the data division method in the previous section, and the session vector output by the behavior coding layer is directly input into the discrimination and execution module to test the adaptability of the control results in the real teaching business link. The test samples cover many kinds of requests, such as course homepage access, material download, assignment submission, score query, discussion board interaction and management configuration call. It includes not only high-frequency and low sensitive operations, but also low frequency and high constraint access. In order to ensure that the evaluation results can reflect the discrimination accuracy and the execution cost at the same time, the experiment is compared from five dimensions of access level identification accuracy, F1 value, unauthorized misjudgment rate, average response delay and blocking hit rate, and four groups of control models are set up: the proposed method, RBAC, ABAC and BiGRU-Control. RBAC is used to test the applicable boundary of static role rules in teaching systems, ABAC

is used to compare the dynamic expression ability of attribute constraint methods, and BirRU-Control is used to compare the performance of pure sequential modeling in Control tasks.

In order to observe the output results of different methods in the overall control task, the experiment first counted the comprehensive indicators of the four types of models on the test set, and the results are shown in Table 5. The accuracy of access level identification reaches 97.3%, the F1 value reaches 96.1%, the unauthorized misjudgment rate is reduced to 0.18%, the average response delay remains at 21 ms, and the blocking hit rate reaches 94.7%. RBAC has the shortest average response time, but the recognition accuracy is low in high-sensitive resources and cross-course jump scenarios, which indicates that it is difficult to cover the context changes in real access links by relying on role boundaries alone. ABAC outperforms RBAC in dynamic constraint handling, but its recognition stability for short-time dense accesses and multi-step jumps is still limited due to the lack of continuous behavior representation. BiGRU-Control can use order information to improve the discrimination results, but there is still a significant accumulation of misjudgments in high-privilege switching scenarios.

Table 5: Comparison of comprehensive effects of different access control methods

Method	Accuracy (%)	F1	Unauthorized Misjudgment Rate	Average Latency (ms)	Blocking Hit Rate (%)
Proposed Method	97.3	0.961	0.18	21	94.7
RBAC	89.6	0.882	0.74	13	81.5
ABAC	92.4	0.907	0.52	18	86.9
BiGRU-Control	94.1	0.928	0.39	24	90.8

As shown in Fig. 7, the proposed method forms clear boundaries of three types of access states. The proportion of passing state samples was 68.4%, and the average intra-class distance was 0.41. Restricted states accounted for 19.7%, and the average intra-class distance was 0.53. The blocking state accounted for 11.9%, and the average intra-class distance was 0.47. The average inter-class distance between the centers of the three types of states is 1.36, which is higher than 0.98 of BiGRU-Control and 0.84 of ABAC. Further statistics show that the accuracy of state recognition is 98.1%, the restricted state is 95.6%, and the blocked state is 94.7%. The overlap between restriction and blocking is only 2.3%, while RBAC reaches 7.8%. This shows that the proposed method can distinguish the three types of access states: normal access, additional verification access and denied access, and maintain a low false positive rate of unauthorized access.

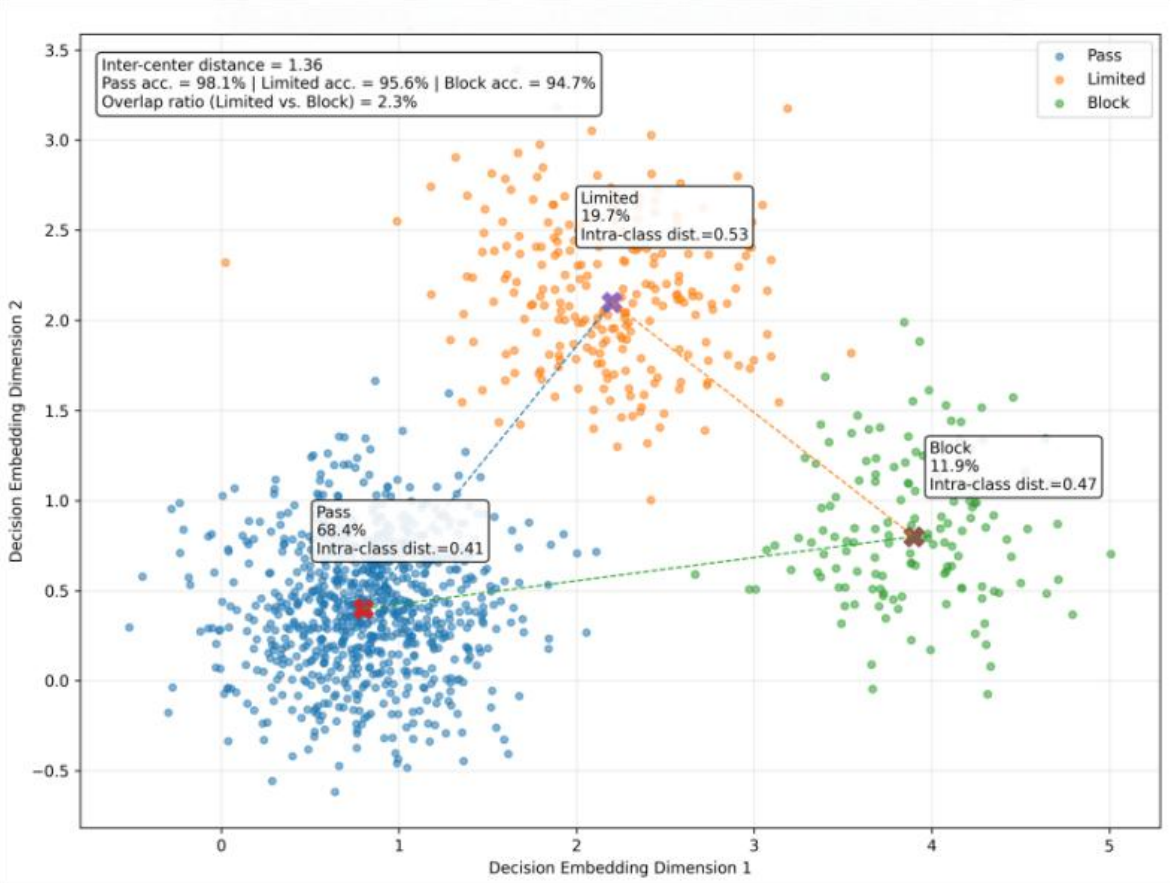

Figure 7: The discriminative spatial distribution diagram of the three types of access states: through, restricted and blocked

In addition to the overall comparison, the experiment continues to analyze the influence of the internal components of the model on the control effect. To this end, ablation experiments are set up in this section to remove time constraint branches, resource relationship branches, risk score branches, and feedback update branches, respectively, and compare the changes of each configuration on the same test set. The results are shown in Table 6. After removing the time constraint branch, the accuracy drops to 95.8%, indicating that the request time interval and phase rhythm have a direct role in distinguishing normal accesses from short-time dense jumps. The most significant decrease in F1 was observed after removing the resource relation branch, indicating that the structural links between the course home page, the resource directory, and the assignment entry support the discriminant boundary. After removing the risk score branch, the unauthorized misjudgment rate increases to 0.31%, indicating that it is difficult to completely describe the abnormal shift on highly sensitive resources by only relying on the permission response vector. After removing the feedback update branch, the average response delay increases and the blocking hit rate decreases, which indicates that the online writeback mechanism has a continuous correction effect on the control stability.

Table 6: Ablation experimental results of the proposed method

Configuration	Accuracy (%)	F1	Unauthorized Misjudgment Rate	Average Latency (ms)	Blocking Hit Rate (%)
Full Model	97.3	0.961	0.18	21	94.7
Without Temporal Constraint Branch	95.8	0.944	0.24	22	92.1
Without Resource Relationship Branch	95.4	0.937	0.26	22	91.6
Without Risk Scoring Branch	96.0	0.948	0.31	20	89.8
Without Feedback Update Branch	95.9	0.945	0.23	25	91.2

As shown in Fig. 8, the four types of resources, including course homepage, assignment entrance, score query and management configuration, show different control trends when the request scale increases from 500 to 4000. The course homepage still has 92.8% pass rate, 5.1% restriction rate, and 2.1% block rate under 4000 requests. The pass rate of the operation entrance decreased to 71.4%, the restriction rate increased to 19.6%, and the blocking rate was 9.0%. The passing rate of score query is 63.7%, the restricted rate is 24.8%, and the blocking rate is 11.5%. The pass rate of management configuration is only 38.6%, the restriction rate is 21.9%, and the block rate is 39.5%. Compared with the 31.2% blocking rate of ABAC in the management configuration scenario and the 18.7% restricted recognition rate of BiGRU-Control in the grade query scenario, the proposed method maintains a clearer control hierarchy for highly sensitive resources.

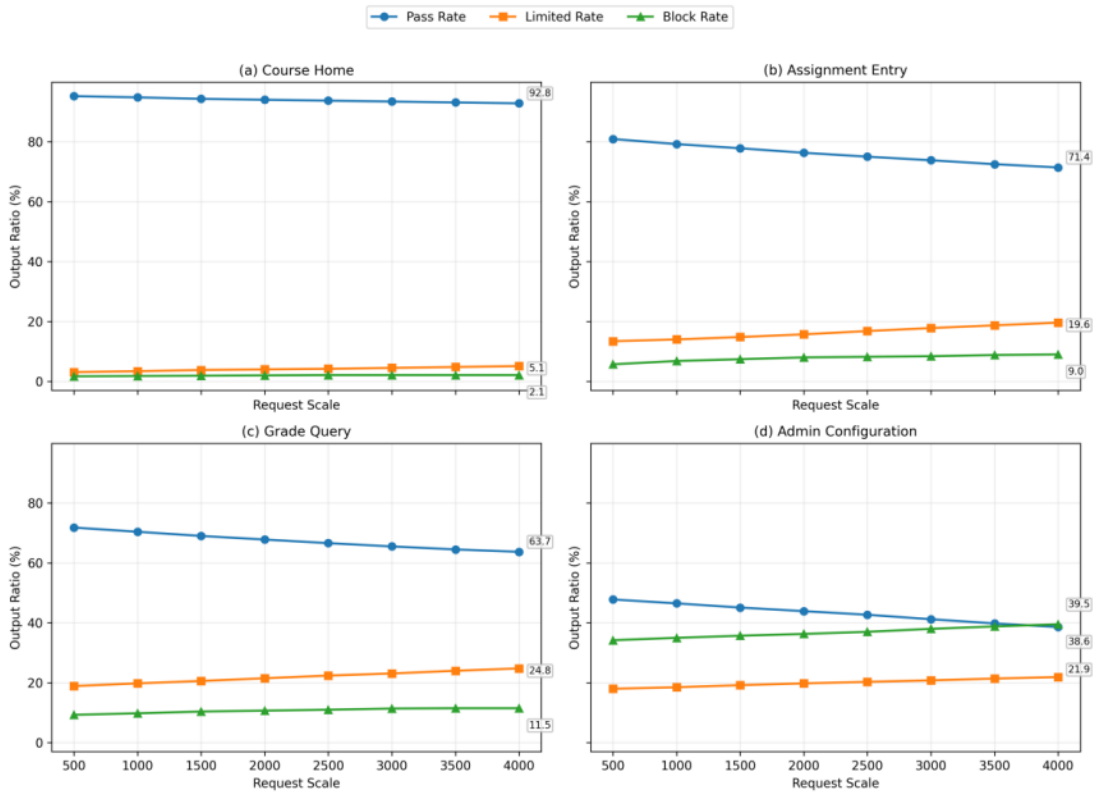

Figure 8: Line chart of access control output variation for four typical types of resources under different request sizes

Synthesizing the above results, it can be seen that the method in this paper has formed a relatively complete discrimination and execution ability in the access control task of university teaching system. The continuous representation provided by the behavior encoding layer enables the control model to identify the state differences in access links. The resource constraints and risk scores together strengthen the boundary expression of highly sensitive requests, and the feedback update mechanism ensures the stability in the continuous control process. The results of comprehensive index comparison, ablation experiment and resource stratification confirm each other, which show that the proposed method can not only improve the accuracy of access level identification, but also reduce the misjudgment rate of unauthorized access while maintaining a low response delay. The obtained control results can directly support three kinds of execution actions in the teaching system, such as resource release, additional verification and blocking audit, and are consistent with the above behavior coding results.

5 Discussion

The method in this paper organizes the behavior coding, permission discrimination and control execution of teaching resources into continuous computing links, and the overall results are consistent with the method design in the previous section. The behavior encoding layer can maintain a more stable representation structure under different teaching stages, and form a clearer constraint response in high-sensitive resource scenarios. In the permission control experiment, the accuracy of access level recognition reaches 97.3%, the F1 value reaches 96.1%, the unauthorized misjudgment rate drops to 0.18%, the average response delay is 21 ms, and the blocking hit rate reaches 94.7%, which shows that the method can stably support the real-time control output in the university teaching system. The four types of resources, including course homepage, assignment entry, score query and management configuration, show obvious stratification under different request scales. Especially in the high-sensitive resource scenario, the proportion of direct release decreases, and the restricted and blocked output is more concentrated, which indicates that the linkage relationship between resource levels, subject differences and operating environment has been effectively included in the control process. The ablation experiments further illustrate that temporal constraints, resource relationships, risk scores, and feedback updates are not additive parts that exist in isolation, but are core components that jointly maintain a stable expression of control boundaries. After removing any branch, the accuracy, blocking effectiveness and response stability will decrease, which indicates that the access control in the university teaching system is not a single point of discrimination task, but a continuous decision-making process supported by multi-dimensional behavioral clues. Therefore, the significance of this method is not only to improve the accuracy of access recognition, but also to enable the teaching platform to complete the joint identification of subject status, resource level and operation context before the request enters the resource layer, so as to provide directly executable calculation basis for release, additional verification and blocking audit.

6 Conclusions

Focusing on the resource access scenario in the university teaching system, this paper proposes a neural network driven teaching resources usage behavior modeling and access control method. The method takes front-end log as input, extracts role, resource, time and path features through behavior sequence coding, and then combines permission constraints,

risk score and feedback update to complete access level discrimination and control execution. The experimental results show that the constructed method can well reflect the permission boundary in real teaching business, and maintain strong consistency in multiple types of resources and multiple access states. The continuous representation formed by the behavior encoding layer provides a stable input for subsequent permission discrimination. The access control layer makes the access control process close to the real business link in the teaching system through the hierarchical output and online writeback mechanism. The method in this paper still has some limitations. The current model mainly relies on a single platform log to complete the training, and the transfer ability in cross-campus, cross-platform and cross-terminal scenarios still needs to be further verified. Although it can maintain a relatively stable control process under the condition of high concurrency, the computational overhead of relationship propagation and feedback update still needs to be compressed when facing larger scale heterogeneous resources. The follow-up research can focus on lightweight deployment, cross-platform transfer learning, incremental update mechanism, low-frequency covert unauthorized identification and interpretable audit interface, so that the model can maintain stronger adaptation and verification capabilities in complex teaching information systems, and enhance the processing ability of transient unauthorized calls, disguised terminal switching and cross-system joint audit scenarios. It also provides more stable technical support for policy maintenance and online correction in actual deployment.

About the Author

Xiao Qian, female, was born in July 1978 in Tianjin. She received her associate degree in International Trade from Nankai University and earned her master's degree in Human Resources from Eastern Michigan University, USA. She is currently affiliated with Cooperative School of International Education at Tianjin University of Commerce. Her research interests focus on educational management.

References

[1] You M, Yin J, Wang H, et al. A knowledge graph empowered online learning framework for access control decision-making[J]. World Wide Web, 2023, 26(2): 827-848.

[2] Shan D, Du X, Wang W, et al. A weighted graphsage-based context-aware approach for big data access control[J]. Big Data, 2024, 12(5): 390-411.

[3] Sun H, Tan Y, Zhu L, et al. A blockchain-based access control protocol for secure resource sharing with mobile edge-cloud collaboration[J]. Journal of Ambient Intelligence and Humanized Computing, 2023, 14(10): 13661-13672.

[4] Roslin Dayana K, Shobha Rani P. Trust aware cryptographic role based access control scheme for secure cloud data storage[J]. Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije, 2023, 64(4): 1072-1079.

[5] Turkea A, Wahid A, Yamani M. Adaptable and Dynamic Access Control Decision-Enforcement Approach Based on Multilayer Hybrid Deep Learning Techniques in BYOD Environment[J]. Computers, Materials, & Continua, 2024, 80(3): 4663.

[6] Shan F, Li F, Wang Z, et al. Deep Learning Social Network Access ControlModel Based on User Preferences[J]. Computer Modeling in Engineering & Sciences (CMES), 2024, 140(1).

[7] Alazab M, Awajan A, Alazzam H, et al. A novel IDS with a dynamic access control algorithm to detect and defend intrusion at IoT nodes[J]. Sensors, 2024, 24(7): 2188.

[8] Liu B, Tang Q. Secure data sharing in federated learning through blockchain-based aggregation[J]. Future Internet, 2024, 16(4): 133.

[9] Pritee Z T, Anik M H, Alam S B, et al. Machine learning and deep learning for user authentication and authorization in cybersecurity: A state-of-the-art review[J]. Computers & Security, 2024, 140: 103747.

[10] Wang C, Tang H, Zhu H, et al. Behavioral authentication for security and safety[J]. Security and Safety, 2024, 3: 2024003.

[11] Budžys A, Kurasova O, Medvedev V. Deep learning-based authentication for insider threat detection in critical infrastructure[J]. Artificial Intelligence Review, 2024, 57(10): 272.

[12] Sağbaş E A, Ballı S. Machine learning-based novel continuous authentication system using soft keyboard typing behavior and motion sensor data[J]. Neural Computing and Applications, 2024, 36(10): 5433-5445.

[13] Wyciślik Ł, Wylężek P, Momot A. The improved biometric identification of keystroke dynamics based on deep learning approaches[J]. Sensors, 2024, 24(12): 3763.

[14] Vegas J, Rao A R, Llamas C. Deep Learning System for User Identification Using Sensors on Doorknobs[J]. Sensors, 2024, 24(15): 5072.

[15] Tao X, Yu Y, Fu L, et al. An insider user authentication method based on improved temporal convolutional network[J]. High-Confidence Computing, 2023, 3(4): 100169.

[16] Bin Sarhan B, Altwaijry N. Insider threat detection using machine learning approach[J]. Applied Sciences, 2022, 13(1): 259.

[17] Roy K C, Chen G. GraphCH: A deep framework for assessing cyber-human aspects in insider threat detection[J]. IEEE Transactions on Dependable and Secure Computing, 2024, 21(5): 4495-4509.

[18] Gong Y, Cui S, Liu S, et al. Graph-based insider threat detection: A survey[J]. Computer Networks, 2024, 254: 110757.

[19] Xiao J, Yang L, Zhong F, et al. Robust anomaly-based insider threat detection using graph neural network[J]. IEEE Transactions on Network and Service Management, 2022, 20(3): 3717-3733.

[20] Villarreal-Vasquez M, Modelo-Howard G, Dube S, et al. Hunting for insider threats using LSTM-based anomaly detection[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 20(1): 451-462.