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

Student performance prediction model in rail transit Teaching based on machine learning

Hongyan Wang[1,*]

[1] Xi'an Traffic Engineering Institute, Meibei W Rd., Huyi District, Xi'an, Shaanxi, 710000

SUMMARY: *In order to support the student performance evaluation in rail transit teaching, this paper constructed a student performance prediction model using multi-source teaching data. The dataset contains 5240 valid samples from the theoretical course platform, driving simulation system, scheduling training terminal, signal disposal module and stage assessment records, and the behavior and evaluation characteristics are established around learning time, operation accuracy, response delay and alarm disposal. After label construction, normalization and feature selection, the model combines behavior weighted mapping, weekly scale aggregation, stability index and structure score with gradient boosting training and temperature calibration to output three types of results and corresponding confidence levels: excellent, standard and early warning. Experimental results show that the accuracy of the proposed model on the test set reaches 94.6%, the macro-average F1 reaches 93.8%, the AUC reaches 0.962, and the ECE is reduced to 0.028. The model maintains stable performance under different modules and repeated tests, which can provide a computable basis for hierarchical identification, process tracking and teaching feedback in rail transit teaching.*

KEYWORDS: *Machine learning, rail transit teaching, student performance prediction, learning behavior modeling*

1 Introduction

Rail transit teaching is characterized by the collaborative promotion of theoretical knowledge, simulation training, fault disposal and safety specifications. The teaching platform, train driving simulator, scheduling training system and online evaluation module will continuously generate behavior logs, operation sequences, performance records and process evaluation data. This kind of data provides a stable data basis for student performance prediction, and also enables machine learning methods to enter the teaching evaluation process of rail transit. Student performance prediction is not a simple estimation of the results of an examination, but a joint calculation of learning progress, training completion, operational stability and stage attainment. Aiming at this goal, this paper takes multi-source teaching data as input, and constructs a prediction link composed of sample sorting, feature modeling, model training and result output, so that the theoretical learning records and practical training process information can be associated in the same computing framework. The writing method not only maintains the professionalism of rail transit teaching scene, but also highlights the technical main line of data processing, feature representation and predictive modeling in the computer direction.

Yağcı studied academic performance prediction in educational data mining and used

*wanghongyan_0131@sina.com
https://doi.org/10.65102/is2026298
Published: 30 April, 2026

machine learning algorithms to extract features from learning data that can be used for classification [1]. Chen H C et al. studied the early prediction of weekly performance in virtual learning environment, and proposed a deep interpretable artificial intelligence framework to make the prediction results on time stage more clearly interpretable [2]. Aljaloud A S et al. studied the prediction of learning results in learning management system and proposed a deep learning model combining convolutional neural network and long short-term memory network [3]. Jawad K et al. studied the prediction of academic performance and participation in virtual learning environment, and proposed a random forest method combined with data balance strategy [4]. Liu T et al. studied the identification of high-risk students and proposed a prediction path based on learning behavior [5]. Li M et al. studied graph structure modeling in student performance prediction and proposed Study-GNN, a multi-topology graph neural network pipeline [6]. Kaur H et al. studied online student academic performance prediction and constructed a machine learning prediction model [7]. Alija S et al. studied the performance prediction under the condition of class imbalance, and proposed a method combining supervised learning and wrapper feature selection [8]. Chen Y et al. studied the expression of temporal dependence in student performance prediction and proposed a prediction model based on self-attention mechanism [9].

Existing research has proved that machine learning methods can extract stable prediction information from teaching data, but there are obvious differences between the data organization mode of rail transit teaching and general online courses. Theoretical learning records, driving simulation trajectories, dispatching and disposal sequences, signal operation logs and stage assessment results are not consistent in sampling frequency, time granularity and semantic level. Only when cross-module alignment, unified coding and label mapping are completed, the model can accurately represent the training status and ability changes of students. Based on this scene feature, this paper models from two levels. First, a multi-source sample organization mechanism for rail transit teaching is established, and time synchronization, feature selection and performance label definition are carried out for classroom records, platform logs, simulation operation and assessment data. The second is to construct a machine learning computing framework for student performance prediction, complete the performance level discrimination and training achievement state output in a unified feature space. This study transformed the professional task process in rail transit teaching into a computable representation, extended the prediction basis from a single score to the whole process behavior sequence, and provided quantitative support for teaching evaluation, hierarchical training and practical training resource allocation.

2 Related work

With the continuous deployment of rail transit teaching platform, driving simulation system, scheduling training software and online evaluation module, the click behavior, task completion sequence, response time, error correction record and stage score of students in the process of learning and training are continuously saved, which provides a computable data basis for performance prediction research. Around this kind of data, student performance prediction has gradually shifted from static classification to time series modeling, behavior fusion and cross-stage evaluation. Related research has formed a relatively clear technical context in the way of feature organization, model structure design and result interpretation.

Nayak et al. studied the academic prediction of students in the educational data mining scenario, and proposed a data mining method based on machine learning classification model to improve the recognition ability of learning performance [10]. Liu Y et al. studied the

connection between online behavior and learning performance during the epidemic, and proposed an empirical analysis path based on online behavior to characterize the impact of learning process on results [11]. Wen X and Juan H studied the early prediction task in online learning activity sequences and proposed a stage prediction method based on deep neural networks [12]. Xiao W and Hu J studied the application evolution of artificial neural networks in student performance prediction, and proposed a systematic review framework for model structures and application scenarios [13]. Lee J E et al. studied student performance prediction in online mathematical games and proposed a comparative analysis scheme of various machine learning algorithms [14]. Fazil M et al. studied student performance prediction driven by participation data, and proposed a new deep learning model to enhance the expression ability of behavioral features [15]. Shou Z et al. studied multi-dimensional time series analysis in online learning and proposed a time series data analysis method for student performance prediction [16]. Ren Y and Yu X studied long-term student performance prediction and proposed an adaptive learning ability algorithm to enhance the stability of long-term prediction [17]. Luo Z et al. studied student performance analysis combining time and space multi-dimensional features, and proposed an integrated method for prediction and analysis [18]. Alnasyan B et al. studied deep learning prediction technology in virtual learning environment and proposed a systematic review on student performance prediction methods [19]. Hernandez-Garcia A et al. studied the relationship between LMS interaction and academic performance, and proposed an analysis path based on learning cycle [20].

The above studies show that the current student performance prediction has extended from a single grade input to the joint modeling of behavior logs, activity sequences, spatio-temporal features and engagement signals, and the model selection has also extended from traditional classifiers to deep networks, time series analysis and adaptive algorithms. Compared with general online courses, rail transit teaching emphasizes more on professional task flow such as driving simulation, scheduling collaboration, signal operation and safety disposal. The data not only has time continuity, but also has module coupling and operational constraints. Therefore, the behavior modeling ideas in related work can provide direct reference, but it is still necessary to complete feature reconstruction and label mapping combined with rail transit teaching scenarios. In order to facilitate the centralized comparison of the technical paths, data types and method characteristics of the existing research, the related work is summarized in Table 1 in this paper.

Table 1: Summary of related work

Author	Method	Data Type	Result Characteristics	Main Contribution
Nayak et al. [10]	Machine Learning Classification	Educational behavioral data	Stable predictive performance	Established a prediction pathway for educational data mining
Wen et al. [12]	Deep Neural Network	Activity sequence data	Supports early identification	Strengthened stage-based prediction capability
Fazil et al. [15]	Deep Learning	Engagement data	Improved behavioral representation	Highlighted the value of engagement modeling
Shou et al. [16]	Time Series Analysis	Multidimensional temporal data	Maintains temporal consistency	Improved the time-series prediction framework
Luo et al. [18]	Spatiotemporal Joint Modeling	Temporal and spatial features	Balances prediction and analysis	Enhanced multidimensional feature fusion
Hernández-García et al. [20]	Learning Cycle Analysis	LMS interaction data	Reveals interaction correlations	Clarified the interpretation pathway of interactive behavior

As shown in Table 1, existing research has formed a relatively mature technology accumulation in classification modeling, deep learning, time series analysis and interactive behavior interpretation, but most of these methods are oriented to ordinary online courses or general learning platforms, and the data structure is relatively single. The performance of students in rail transit teaching is not only affected by classroom learning records, but also directly related to simulation operation accuracy, task response sequence, rule execution status and stage assessment results. Therefore, based on the absorbing of existing research and technical ideas, this paper integrates multi-source teaching data organization, training process feature expression and machine learning prediction mechanism into unified computing link. So that the related work can be more targeted extension in the rail transit teaching scene. In this way, the model input is no longer limited to a single performance, but covers the whole process behavior trajectory and training state representation

3 The construction of student performance prediction model for rail transit teaching scenarios

3.1 Multi-source data sorting and sample labeling of rail transit teaching

Student performance prediction for rail transit teaching is not based on linear extrapolation of single evaluation score, but based on the unified organization of multi-source training data. The theory course platform, driving simulation system, scheduling training terminal, signal disposal module and stage assessment system will generate learning records, operational events, response time, disposal results and process scores, respectively. These data have obvious differences in sampling granularity, field semantics and storage structure. Without a unified data sorting and labeling mechanism, it is difficult to form a stable sample expression of the behavior trajectory of the same student in different modules, and it is difficult to maintain a consistent calculation scale of model input. Therefore, this section first establishes the sample organization framework for rail transit teaching scenarios, and converts the heterogeneous records scattered in each business module into comparable, aggregable, and trainable standard data sets, so as to provide reliable input for subsequent machine learning prediction.

As shown in Fig. 1, the data collation process in this paper consists of four levels: source data access, field processing, sample construction, and label generation. The bottom input includes course platform logs, driving simulation records, scheduling operation files, online evaluation results and stage evaluation tables. The middle layer completed identity desensitization, time alignment, field mapping, exception elimination and missing completion. The upper layer aggregates the behavior records from different sources according to "students-training weeks-course modules", and generates comprehensive performance labels based on theoretical scores, practical training performance and process behavior. Through this link, the original records, which were originally scattered in multiple systems, are compressed into training samples with uniform structure.

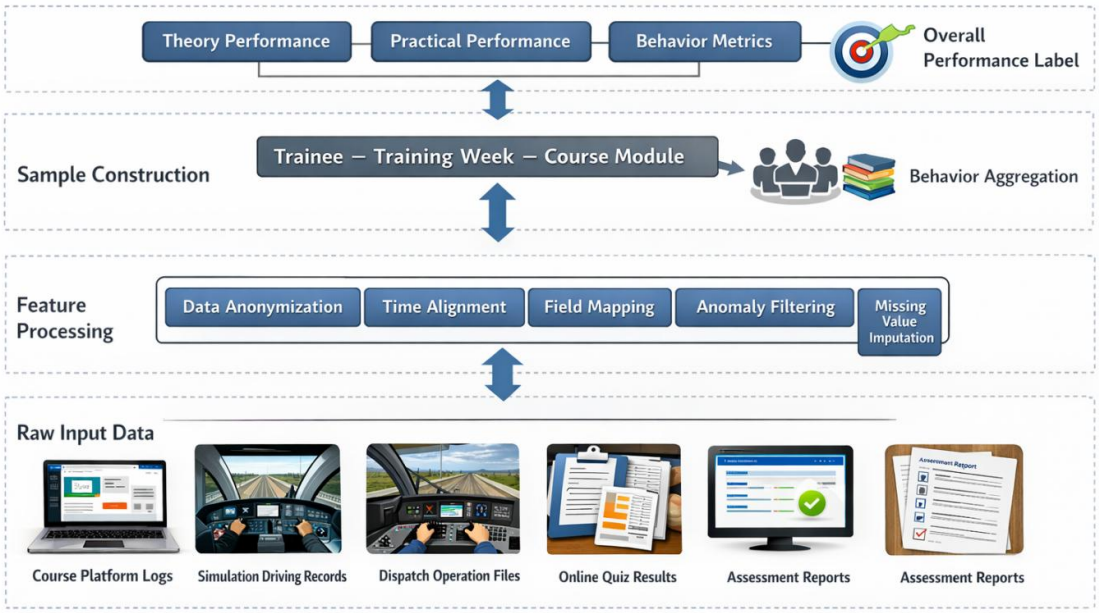

Figure 1: Process of multi-source data sorting and sample labeling for rail transit teaching

In the continuous feature processing stage, this paper adopts the min-max normalization method to unify the scales of various behavior quantities and achievement quantities, and the calculation formula is as follows:

$$x_{ij}^{*} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j) + \varepsilon} \tag{1}$$

where x_{ij} represents the original value of the i student on the j feature, $\min(x_j)$ and $\max(x_j)$ represent the minimum and maximum value of the j feature in all samples, ε represents the smoothing term, and x_{ij}^{*} represents the normalized result. The function of Equation (1) is to eliminate the differences in the numerical range of learning time, operation delay, alarm number and stage performance, so that data from different sources can be mapped into a unified feature space, and then ensure that the subsequent model training keeps the reading of various features consistent.

After cleaning and scale unification, this paper retains the core fields directly related to the training status of students, and establishes a structured sample table accordingly. The structure of the core fields after preprocessing is shown in Table 2.

Table 2: Core field structure after preprocessing

Field Name	Code	Source Module	Data Type	Field Description
Training Week	week	Teaching Platform	Integer	Unified temporal index
Learning Duration	learn_t	Course Platform	Float	Effective learning minutes within the week
Operation Accuracy	op_acc	Driving Simulation	Float	Proportion of standard actions completed
Response Delay	resp_d	Dispatch Training	Float	Average response time for tasks
Alarm Handling Count	alarm_n	Signal Module	Integer	Total number of valid handling actions
Stage Score	score_s	Assessment Module	Float	Score of stage-based evaluation

In the sample labeling stage, this paper does not directly use the single test score as the prediction label, but synthetically combines the theoretical evaluation, practical training performance and process behavior into a comprehensive performance score. The synthetic label is calculated as follows:

$$y_i = \alpha s_i + \beta o_i + \gamma c_i - \lambda r_i, \quad \alpha + \beta + \gamma + \lambda = 1 \tag{2}$$

Among them, y_i represents the comprehensive performance score of the i student, s_i represents the theoretical evaluation score, o_i represents the training operation score, c_i represents the training completion degree, r_i represents the penalty term composed of response delay and operation error, and α, β, γ and λ represent the weight of each. The function of Equation (2) is to combine the result performance and process behavior into the label construction link, so that the sample labeling can reflect the knowledge mastery, operation quality and training stability at the same time.

In summary, this section completes the unified access, field screening, scale standardization, sample merging and label generation of multi-source data of rail transit teaching, so that the original heterogeneous records are transformed from teaching business data into standard sample sets that can be directly called by the model. After this processing, the samples are consistent in time index, field semantics and label definition, which provides a standard data input for the subsequent learning behavior and feature modeling of the training process, and also lays a foundation for the stable training of machine learning prediction models.

3.2 Feature modeling method of learning behavior and training process

The learning behavior in rail transit teaching has obvious characteristics of process and task. The course platform records the progress of knowledge learning and answer activities, the driving simulation system saves the manipulation sequence and action deviation, the scheduling training terminal reflects the order and completion time of the instruction response, and the signal disposal module corresponds to the alarm confirmation, interlock recovery and standard execution state. Although these behavioral data are related to student performance, they are not consistent in sampling mode, time granularity and semantic level, so they cannot be directly input into the prediction model as homogeneous features. In order to make the multi-source behavior data form a comparable, aggregable and discriminative representation in the same computing space, this paper constructs a feature extraction method of learning behavior and training process according to the path of "correlation screening - time series aggregation - stability characterization - density structure enhancement", so as to transform the original teaching behavior sequence into a unified feature set suitable for machine learning modeling.

In order to ensure that the behavioral variables entering the model have a clear statistical correlation with the comprehensive performance of students, this paper first calculates the linear correlation degree between each behavioral characteristic and the comprehensive label, and its expression is shown in Equation (3).

$$r_j = \frac{\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3}$$

Here, r_j represents the correlation coefficient between the j behavior feature and the comprehensive performance label, x_{ij} represents the observation value of the i student on

the j feature, $\bar{x}_j$ represents the mean value of the feature over all samples, y_i represents the comprehensive performance label of the i student, $\bar{y}$ represents the mean value of the comprehensive label of all samples, and n represents the total number of samples. Equation (3) is used to screen out features more closely related to performance results from a large number of original behavioral variables, so that the subsequent modeling process can focus on highly related variables such as learning time, operation accuracy, response delay and disposal completion.

After completing the correlation screening, this paper takes the training week as the time window, and combines the theoretical learning, task execution and process completion scattered in the same week into a unified behavioral intensity index, which is calculated as shown in Equation (4).

$$h_i^{(t)} = \alpha l_i^{(t)} + \beta o_i^{(t)} + \gamma c_i^{(t)}, \quad \alpha + \beta + \gamma = 1 \tag{4}$$

Among them, $h_i^{(t)}$ represents the comprehensive behavior intensity of the i student in week t; $l_i^{(t)}$ represents the learning input of the week, which is usually composed of effective learning time, answer frequency and resource access times; $o_i^{(t)}$ represents the execution quality of training, which mainly reflects the standard completion degree in driving simulation and scheduling operation; $c_i^{(t)}$ represents the task completion degree. Reflect the achievement proportion of training tasks in that week, and α, β, and γ represent the weight coefficients of the three types of behavior subitems. Equation (4) compreszes the discrete behavior within a week into a single intensity value to provide a unified input for subsequent timing analysis.

The intensity value alone is not enough to describe the continuous change of the training state. Therefore, this paper further defines the behavioral stability, which is used to describe the degree of fluctuations of students in consecutive training weeks, and its expression is shown in Equation (5).

$$s_i = 1 - \frac{1}{T-1}\sum_{t=2}^{T}\left|h_i^{(t)} - h_i^{(t-1)}\right| \tag{5}$$

Here, s_i represents the behavioral stability of the i trainee within the whole training phase, T represents the number of training weeks, and $h_i^{(t)}$ and $h_i^{(t-1)}$ represent the behavioral intensity of the two adjacent weeks, respectively. Equation (5) reflects the smoothness of the training state by accumulating the change amplitude between adjacent time Windows, and the higher the stability is, the more continuous the behavior output is and the more balanced the training execution is.

Fig. 2 shows the overall path of learning behavior and training process feature modeling. The left side of the figure is the original input layer, which contains course platform logs, driving simulation sequences, scheduling response records, and signal disposal events. In the middle part, correlation screening, weekly scale aggregation and stability calculation are completed in turn. On the right is the structure enhancement layer, which forms the final feature output through weighted representation, distance calculation and density score.

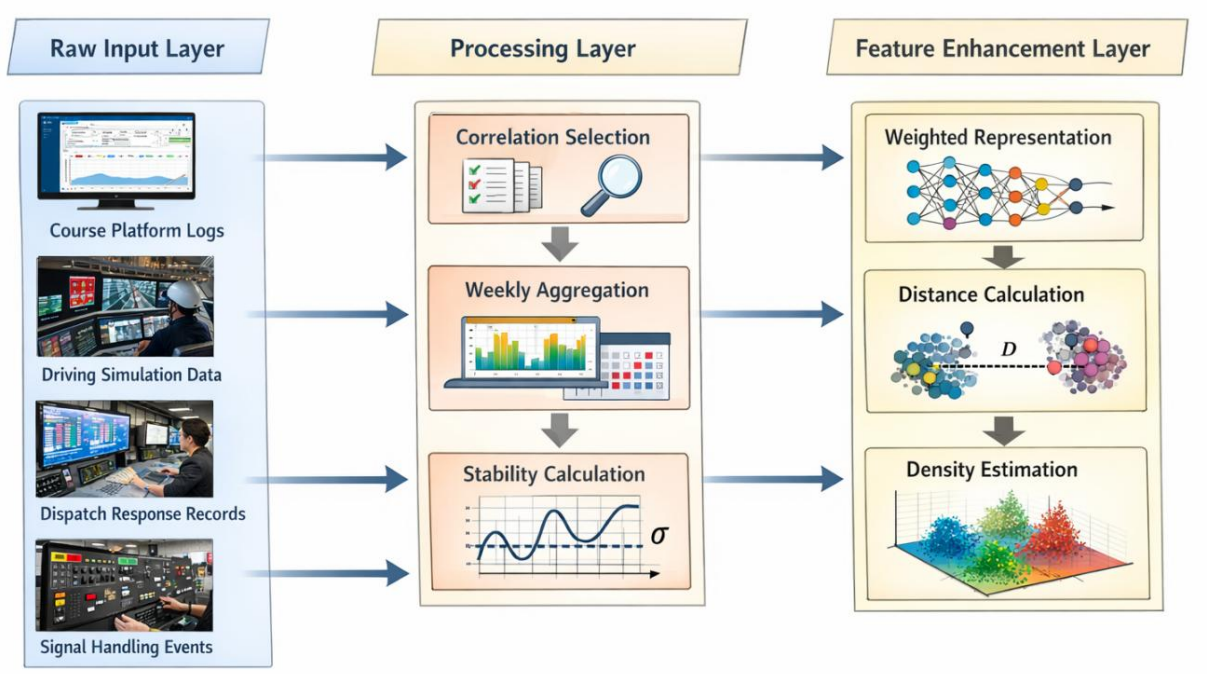

Figure 2: Learning behavior and training process feature modeling process

After obtaining the time series aggregation results, this paper maps the filtered features into weighted representation vectors to enhance the expression strength of highly relevant behavioral variables in the feature space, and its expression is shown in Equation (6).

$$z_i = [r_1 x_{i1}^*, r_2 x_{i2}^*, \ldots, r_m x_{im}^*] \tag{6}$$

Here, z_i represents the weighted feature vector of the ith student, m represents the number of retained features, r_1 represents the correlation coefficient of the first feature, and x_{i1}^* represents the normalized behavior feature value. Equation (6) directly embeds the statistical correlation into the vector construction process, so that the behaviors with high correlation occupy higher weights in the subsequent distance calculation and density analysis.

In the weighted representation space, Euclidean distance is used in this paper to depict the overall behavioral difference between different trainees, which is calculated as shown in Equation (7).

$$d_{ij} = \|z_i - z_j\|_2 \tag{7}$$

Here, d_{ij} represents the feature distance between student i and student j, and z_i and z_j represent the weighted feature vectors of two students, respectively. Equation (7) is used to measure the comprehensive differences of different students in learning engagement, practical training execution and training rhythm, which is the basis for subsequent density construction.

In order to further identify high-density regions in the behavior distribution, this paper constructs a local density index based on the distance calculation, which is defined as shown in Equation (8).

$$\rho_i = \sum_{j \neq i} \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) \tag{8}$$

Here, ρ_i represents the local density around the i trainee sample, d_{ij} represents the inter-sample distance, and σ represents the density smoothing parameter. Equation (8) counts the aggregation degree of neighborhood samples through the distance decay function, which is used to identify the sample area with relatively concentrated behavior patterns.

Under the condition that the local density is known, this paper continues to calculate the minimum distance from the sample to the sample with higher density to judge its central position in the overall distribution, and its expression is shown in Equation (9).

$$\delta_i = \min_{j:\rho_j>\rho_i} d_{ij} \tag{9}$$

Here, δ_i represents the closest distance between sample i and samples with higher density. For the sample with the highest global density, δ_i takes the maximum distance from other samples in the dataset. Equation (9) enables samples with high density and far away from other high-density clusters to be identified as more representative structural centers.

On this basis, the local density and center distance are combined as the final structure score, which is used to represent the representativity of the sample in the behavior distribution. The calculation method is shown in Equation (10).

$$q_i = \rho_i \cdot \delta_i \tag{10}$$

Here, q_i denotes the comprehensive structure score of the i trainee sample, ρ_i reflects the strength of aggregation around the sample, and δ_i reflects the degree of separation of the sample with respect to higher density regions. Equation (10) is able to preserve both aggregation and centrality information, so that the subsequent prediction model can not only see a single feature value, but also utilize the structural position of the sample in the overall behavior space.

In summary, the feature extraction of learning behavior does not stay at the simple statistical level, but determines the effective variables through correlation screening, obtains the unified behavior strength through weekly scale aggregation, describes the training continuity through stability calculation, and then enhances the feature expression through weighted representation, distance measurement and density structure score. After this process, the original behavior sequence in rail transit teaching is transformed into a feature set with statistical correlation, time evolution and distribution structure information, which provides a more solid input basis for subsequent machine learning prediction models.

3.3 Machine learning prediction model training and result output mechanism

After completing the feature modeling of learning behavior and training process, the task of model training phase is no longer to simply receive a number of statistics, but to organize the formed weighted behavior vector, training stability and structure score into a unified input, and on this basis to complete multi-classification learning, probability calibration and result output. The samples in the rail transit teaching scene contain not only the static features formed by curriculum learning, but also the process features formed by driving simulation, scheduling operation and signal disposal. Therefore, the predictor must not only have the ability to express the combination of nonlinear features, but also be able to output the level results with interpretability. Based on this consideration, this paper uses the gradient boosting tree model as the core trainer, and adds a temperature calibration mechanism in the output layer, so that the model can not only complete the classification decision, but also give the confidence information of the boundary samples.

In order to ensure that features from different sources use the same input interface in the training phase, this paper first integrates the behavior vector, stability index and structure score obtained in the previous section into a unified input vector, which is defined as shown in Equation (11).

$$u_i = [z_i, s_i, q_i] \tag{11}$$

where u_i represents the final input vector corresponding to the i student; z_i represents the behavior feature vector obtained by correlation screening and weighted mapping, which is used to characterize learning engagement, training quality and task execution characteristics. s_i represents the training stability, which reflects the fluctuation between consecutive training weeks. q_i stands for structure score and is used to describe how representative the sample is in the overall behavior space. The function of Equation (11) is to compress the statistical features, temporal features and structural features into the same input layer, so that the model training does not rely on scattered multiple channels, but complete the subsequent iterations in the form of unified samples.

In the model training phase, this paper uses the multi-class lifting structure to calculate the cumulative output score of each category, and its expression is shown in Equation (12).

$$o_{ic} = \sum_{m=1}^{M} f_{mn,c}(u_i) \tag{12}$$

where o_{ic} represents the cumulative output score of sample i on class c; M represents the total number of base learners participating in the iteration; $f_{mn,c}(u_i)$ is the output of the m tree for class c. Equation (12) shows that the final category score is not given by a single learner at once, but is formed by stacking multiple trees round by round. Such a structure can continuously correct the residual of the previous round, so that the model can maintain a higher identification ability for the complex behavior interaction in the rail transit teaching samples.

In order to control the training direction and suppress the unstable fitting caused by unconstrained growth, this paper constructs the overall objective function with a regularization term, whose expression is shown in Equation (13).

$$\mathcal{L} = \sum_{i=1}^{n} \ell(y_i, o_i) + \sum_{m=1}^{M} \Omega(f_m) \tag{13}$$

Here $\mathcal{L}$ represents the overall objective function in the training phase; $\ell(y_i, o_i)$ is the loss term for the i sample, which measures the deviation between the true label y_i and the predicted output o_i. Let $\Omega(f_m)$ denote the complexity term of the m base learner. The function of Equation (13) is to bring "classification accuracy" and "structural constraints" into the same optimization framework, so that the model training not only pursues better prediction results, but also maintains a reasonable model scale.

In the above objective function, the complexity of each base learner is characterized by Equation (14).

$$\Omega(f_m) = \gamma T_m + \frac{1}{2}\lambda \sum_{j=1}^{T_m} w_j^2 \tag{14}$$

where T_m represents the number of leaf nodes in the m tree; w_j represents the leaf value weight corresponding to the j leaf node; γ is the structure penalty coefficient, which is used to control the complexity growth caused by new leaf nodes. λ denotes the leaf-valued regularization coefficient, which is used to limit the leaf weight to be too large. By simultaneously limiting the "number of nodes" and "leaf value amplitude", Equation (14) makes the model not form too fine split due to local noise when facing the rail transit teaching samples, so as to maintain a more stable generalization ability.

In the process of tree structure generation, the most critical step is to find the optimal splitting point. To this end, in this paper, a gain function is used to evaluate the effectiveness of the candidate splitting, which is calculated as shown in Equation (15).

$$\text{Gain} = \frac{1}{2}\left(\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda}\right) - \gamma \tag{15}$$

where G_L and G_R represent the sum of the first degree of the left child node and the right child node respectively, and G represents the sum of the first degree of the parent node. H_L and H_R denote the second-order gradient sum of the left and right child nodes, respectively, and H denotes the second-order gradient sum of the parent node. The λ and γ are kept consistent with the regularization parameters in Equation (14). The role of Equation (15) is to quantify "whether the current split is worth preserving", and the split is accepted only if the gain brought by the left and right children can offset the structural penalty. The tree structure thus formed is more in line with the true discriminative boundary in the behavioral features.

The overall process of machine learning prediction model training and result output is shown in Fig. 3. The leftmost part of the figure is the input layer, which is composed of weighted behavioral features, training stability, and structure score. The middle part is the training layer, which completes the objective function optimization, tree structure splitting and multiple rounds of iterative accumulation in turn. The output layer, on the far right, converts the class scores into probabilities, then calibrates them to a smoother posterior distribution, and finally outputs the excellent, qualified, and early warning performance levels and their confidence levels.

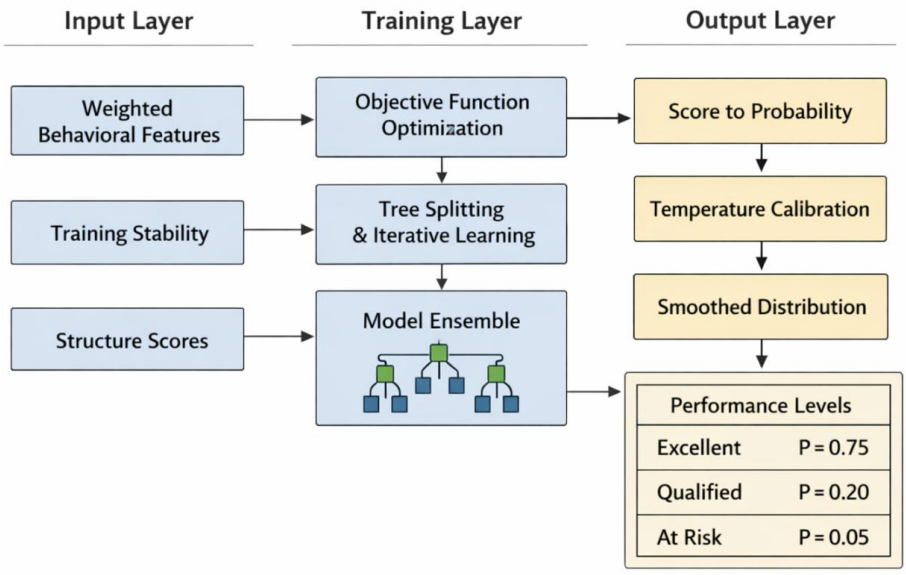

Figure 3: Machine learning prediction model training and result output mechanism

When the cumulative scores of multiple trees are calculated, the Softmax function is used in this paper to map the category scores into an initial probability distribution, whose expression is shown in Equation (16).

$$p_{ic} = \frac{\exp(o_{ic})}{\sum_{k=1}^{C} \exp(o_{ik})} \tag{16}$$

Here, p_{ic} represents the original predicted probability that sample i belongs to class c. o_{ic} represents the cumulative score of sample i on class c; C represents the total number of categories. The function of Equation (16) is to convert the score values on different categories into comparable probability quantities, so that the model output no longer stays at the abstract score level, but enters the probability space that can be interpreted.

Considering that the boundary samples may still have over-steep distribution under the original Softmax output, this paper further introduces the temperature calibration mechanism, whose expression is shown in Equation (17).

$$\tilde{p}_{ic} = \frac{\exp(o_{ic}/\tau)}{\sum_{k=1}^{C} \exp(o_{ik}/\tau)} \tag{17}$$

Here, $\tilde{p}_{ic}$ represents the calibrated class probability; When τ increases, the output distribution will be smoother. o_{ik} represents the uncalibrated category score. The function of Equation (17) is to reduce the excessive category bias on the boundary samples, make the model output maintain a more reasonable relative relationship between different categories, and thus improve the reliability of the interpretation of the results.

In the final result output stage, this paper completes the grade judgment and confidence output based on the calibrated probability, which is defined as shown in Equation (18).

$$\hat{y}_i = \arg\max_c \tilde{p}_{ic}, \quad \kappa_i = \max_c \tilde{p}_{ic} \tag{18}$$

Where $\hat{y}_i$ represents the final predicted class of sample i; Let κ_i denote the maximum calibration probability over the corresponding class, which is the confidence of the current decision. The function of Equation (18) is not only to give the final classification results, but also to provide teachers with a quantitative basis for "determining whether it is stable". For the samples with low κ_i, even if the class has been determined, they can still be regarded as boundary samples that need further observation.

In summary, this paper completes the complete modeling process from unified input construction, objective function establishment, tree structure training, probability mapping to rank output in this section. Compared with the writing method that only outputs a single classification label, this mechanism integrates training constraints, probability calibration and confidence output into the prediction link at the same time, so that the multi-source behavior features in rail transit teaching can be stably transformed into performance prediction results that are interpretable, comparable and can be used for teaching feedback.

4 Model experiment and performance analysis for rail transit teaching

4.1 Experimental environment and training parameter configuration

In order to verify the training stability, parameter adaptation and cross-module generalization

ability of the constructed rail transit teaching student performance prediction model under unified computing conditions, this paper configured the system around the experimental platform, sample organization, comparison model and parameter optimization process. The experimental platform uses Windows 11 64-bit operating system, Intel Core i9 processor, 64 GB memory, Python 3.10 programming language, and PyCharm development environment. The model training is implemented by PyTorch and Scikit-learn, and the sample storage and invocation are completed by MySQL 8.0. This environment configuration can meet the computational requirements of reading and writing multi-source teaching data, feature vector construction, model iterative training and parallel evaluation of validation set, and also ensure that the subsequent comparison experiments are carried out under the same hardware conditions.

The experimental data come from the theoretical course platform, driving simulation system, scheduling training terminal, signal disposal module and stage assessment records in rail transit teaching scenarios from 2019 to 2024, and a total of 5240 valid samples are sorted out. The sample takes "students-training weeks-course module" as the basic index unit, and is divided into training set and test set according to the proportion of 80% and 20%. In order to reduce the influence of single division on the results, five-fold cross validation is introduced in the training phase, and the validation set is used to complete the parameter search and output calibration. In this paper, the proposed Model is denoted as Model-A, and compared with the random forest model, support vector machine model and multilayer perceptron model, in order to observe the adaptation differences of different types of learners on rail transit teaching data. To facilitate the illustration of the experimental platform and the core parameter configuration, the relevant Settings are shown in Table 3.

Table 3: Experimental environment and core parameter Settings

Item	Configuration
Operating System	Windows 11 64-bit
CPU and Memory	Intel Core i9, 64 GB RAM
Development Environment	Python 3.10, PyCharm
Training Framework	PyTorch, Scikit-learn
Database	MySQL 8.0
Sample Size	5,240
Data Split	80% training set, 20% test set
Cross-Validation	10-fold
Number of Trees	300
Learning Rate	0.05
Maximum Depth	6
Temperature Coefficient	1.6

In the parameter optimization process, the number of trees is increased from 100 to 400, the learning rate is searched in the range of 0.01 to 0.10, the maximum depth is adjusted in the range of 4 to 8, the minimum number of leaf node samples is compared in three sets of 10, 20, and 30, and the temperature coefficient is jointly determined based on the negative log-likelihood and calibration error on the validation set. Experimental results show that when the number of trees is 300, the learning rate is 0.05, the maximum depth is 6, the minimum number of leaf node samples is 20, and the temperature coefficient is 1.6, the AUC, Macro-F1 and calibration consistency on the validation set maintain a high level, and the training rounds and convergence speed are more balanced. Based on this set of parameters, Model-A has smoother output probabilities and less fluctuation in the decision of boundary samples while

maintaining the classification ability. The experimental configuration thus determined not only ensures the repeatability of model training, but also provides a unified and stable calculation basis for subsequent performance evaluation and comparative analysis.

4.2 Prediction results and model effect evaluation

In order to test the training efficiency of the proposed model in rail transit teaching scenarios, this paper first compares the convergence process of the four types of models before and after preprocessing. As shown in Fig. 4, Model-A enters the stable interval around the 42nd round after the unified sample consolidation is completed, while it takes about 71 rounds to achieve similar accuracy without the standardized consolidation, and the convergence rounds are reduced by 29. Under the same conditions, Model-B is reduced from 83 rounds to 63 rounds, Model-C from 96 rounds to 74 rounds, and Model-D from 108 rounds to 88 rounds. It can be seen that after unified coding, time alignment and label normalization of multi-source records in rail transit teaching, the overall training efficiency is improved, but the convergence benefit of Model-A is the most obvious, indicating that the input organization method proposed in this paper has higher consistency between the predictor structure.

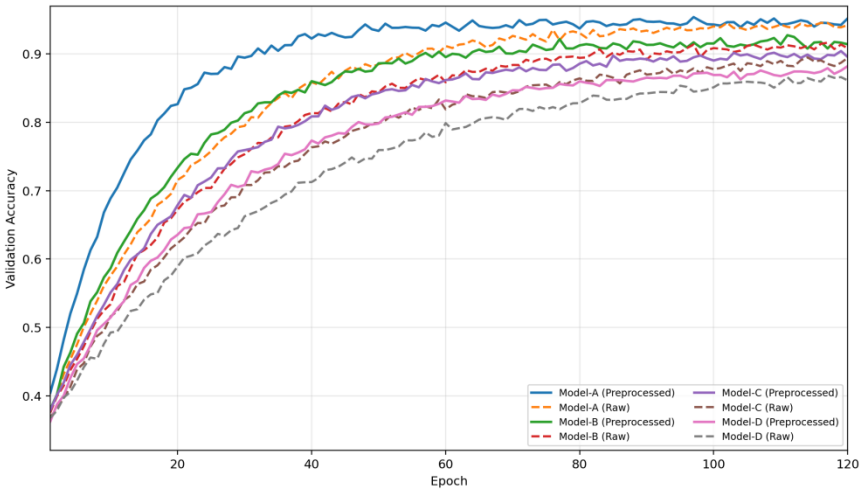

Figure 4: Comparison of the training convergence process of the four classes of models before and after preprocessing

In order to investigate the ability of the model to describe the intrinsic structure of the training samples, this paper further compares the fitting effect. As shown in Fig. 5, the fitting correlation coefficient of Model-A on the training samples reaches 0.961, which is significantly higher than 0.928 of Model-B, 0.903 of Model-C and 0.887 of Model-D. After observing the fitting curve of the high segment samples and the boundary segment samples, it can be found that the deviation of Model-A in the local fluctuation interval is smaller. Especially in the section where the driving simulation score and the scheduling response score change synchronously, the predicted curve keeps high consistency with the real label. This shows that the feature set composed of weighted behavior vector, stability index and structure score can more fully express the composite training state in rail transit teaching scenarios.

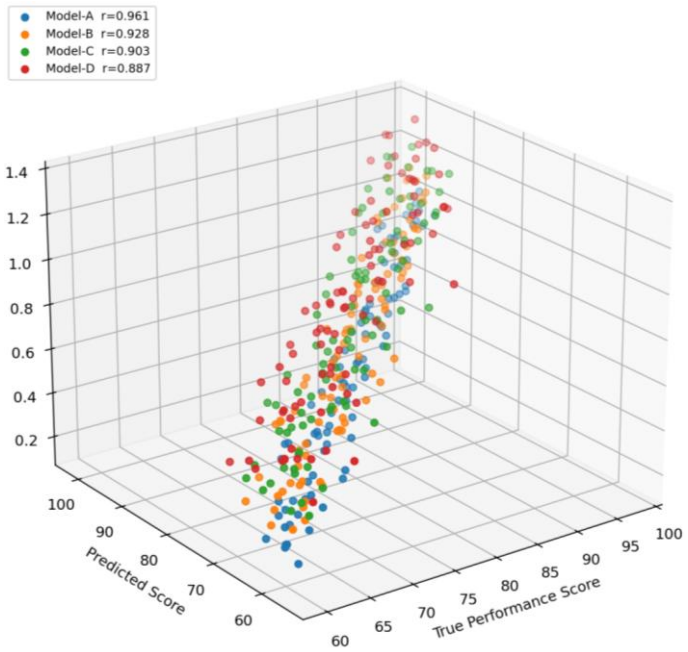

Figure 5: Comparison of the fitting effect of the four classes of models on the training samples

In the test set performance evaluation, accuracy, Macro-F1, AUC, ECE, and 10-fold cross validation results are counted as independent experiments, and the results are shown in Table 4. The accuracy of Model-A is 94.6%, Macro-F1 is 93.8%, AUC is 0.962, ECE is only 0.028, ten-fold cross validation average accuracy is 93.9%, and standard deviation is 0.47%. In comparison, the accuracy and Macro-F1 are 91.7% and 90.9% for Model-B, 89.8% and 88.6% for Model-C, and 87.9% and 86.8% for Model-D. From the table, we can see that Model-A not only keeps the lead in classification accuracy, but also is more stable in probability consistency and interfold fluctuation control, which indicates that the Model has stronger generalization ability on rail transit teaching samples.

Table 4: Overall performance comparison of the four classes of models on the test set

Model	Accuracy (%)	Macro-F1 (%)	AUC	ECE	Average Accuracy over 10 Folds (%)	Standard Deviation over 10 Folds (%)
Model-A	94.6	93.8	0.962	0.028	93.9	0.47
Model-B	91.7	90.9	0.936	0.041	91.2	0.56
Model-C	89.8	88.6	0.918	0.052	89.4	0.71
Model-D	87.9	86.8	0.901	0.064	87.6	0.83

In order to further test the deviation stability between the output results and the real performance, this paper performed 60 repeated tests on the four types of models, and counted the change of the error rate of each test. As shown in Fig. 6, the error rate of Model-A is mainly distributed between 0.03 and 0.09, and the average error rate is 0.052. The average error rate of Model-B is 0.073. Model-C is 0.089; The Model-D value is 0.108. Model-A not only has the lowest average error, but also has the narrowest-fluctuation bandwidth, which indicates that the prediction output still has high consistency in the presence of sample perturbation, training partition change, and local label fluctuation. This result mutually confirms the previous AUC and Macro-F1 performance.

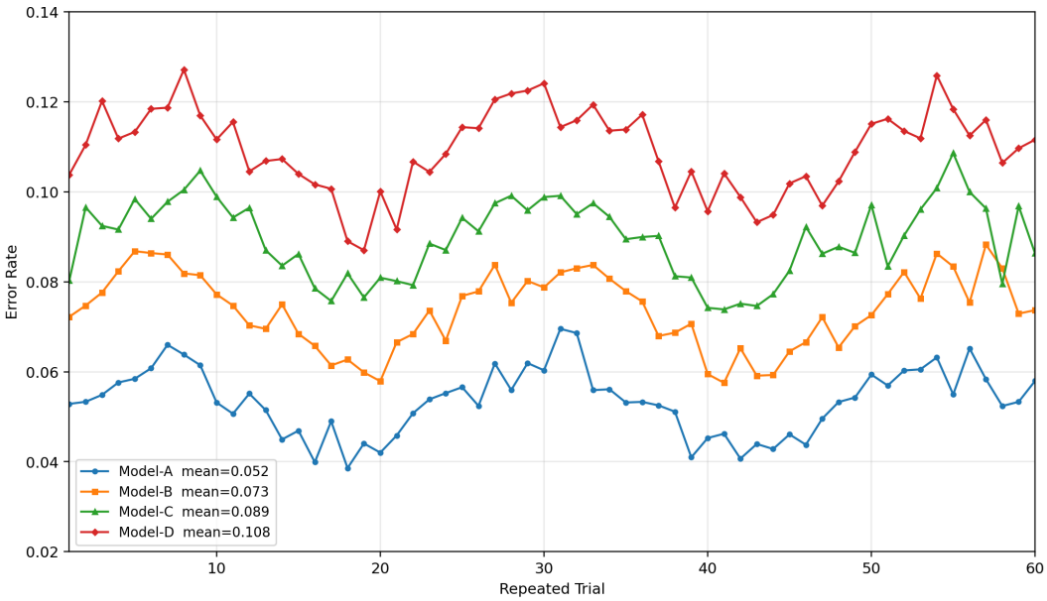

Figure 6: Variation in error rates for the four classes of models over 60 repeated trials

In addition to the repeated trial error, this paper separately compares the accuracy fluctuation of the four types of models in 10-fold cross-validation. As shown in Fig. 7, the 10 discount results of Model-A are concentrated between 93.2% and 94.5%, and the discount difference is controlled within 1.3 percentage points. Model-B fluctuates from 90.6% to 91.8%; 88.7% to 89.9% for Model-C; For Model-D, it was 86.9% to 88.1%. From the fluctuation range, it can be seen that the performance of Model-A is most concentrated on different data folds, indicating that it is less sensitive to the change of training sample distribution, and is more suitable for the prediction task under multi-module, multi-stage and asynchronous sampling conditions in rail transit teaching.

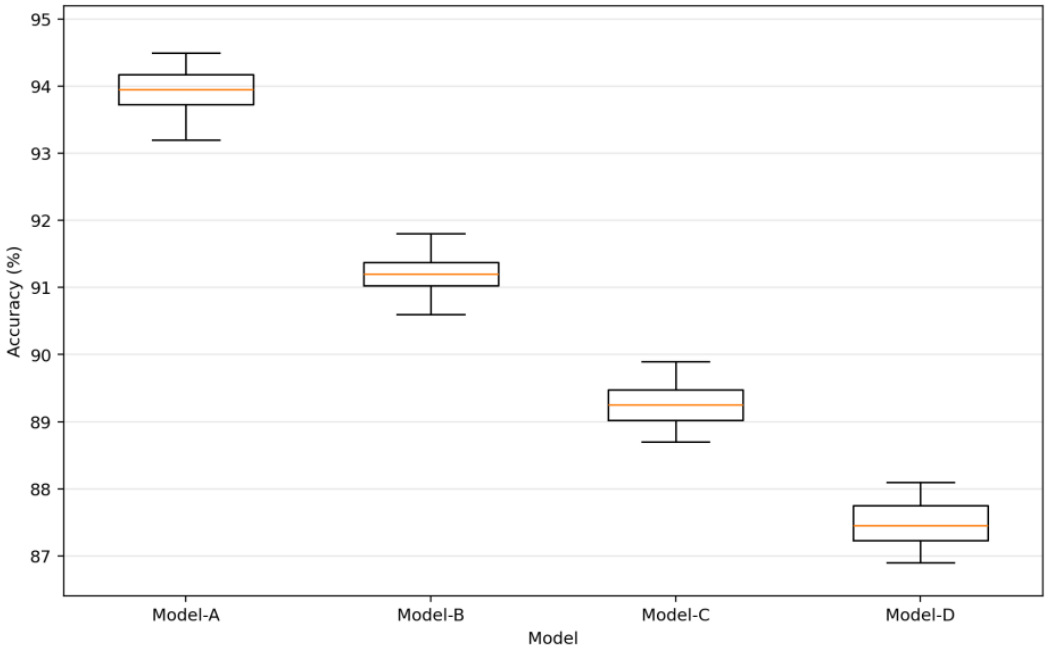

Figure 7: Accuracy fluctuation comparison of four classes of models in 10-fold cross-validation

After the performance of the main model is clear, this paper continues to conduct ablation experiments to examine the contribution of each component module to the overall results, and the results are shown in Table 5. The Accuracy of full Model-A is 94.6%, Macro-F1 is 93.8%, and AUC is 0.962. After removing the stability index, the Accuracy decreased to 92.7%, and Macro-F1 decreased to 91.6%. After removing the structure score, the Accuracy further drops to 92.1%, indicating that the density structure information has an obvious supporting effect on sample discrimination. After removing the temperature calibration, AUC only decreases from 0.962 to 0.956, but ECE increases from 0.028 to 0.067, indicating that this module has the greatest impact on the consistency of probability interpretation. After removing the behavioral weighted mapping, the Accuracy is reduced to 91.8%, and Macro-F1 is reduced to 90.5%, which shows that the correlation screening and weight fusion are the key links of the input expression of the whole model.

Table 5: Results of ablation experiments for Model-A

Model Variant	Accuracy (%)	Macro-F1 (%)	AUC	ECE
Full Model-A	94.6	93.8	0.962	0.028
Without Stability Indicator	92.7	91.6	0.941	0.041
Without Structural Scoring	92.1	90.8	0.936	0.044
Without Temperature Calibration	93.4	92.3	0.956	0.067
Without Behavioral Weighted Mapping	91.8	90.5	0.932	0.049

In order to further investigate the transfer and adaptation ability of the model on different teaching modules, this paper divides the test samples into four subsets according to the course attributes: theoretical learning, driving simulation, scheduling training and signal processing, and statistics the Accuracy and Macro-F1 of the four types of models on each subset. This experiment no longer focuses on the overall average performance, but examines the ability of the model to maintain in the face of different task types. The results are shown in Table 6. It can be seen that Model-A keeps leading in all four modules, where the Accuracy on the driving simulation subset reaches 95.1% and Macro-F1 reaches 94.4%. It achieves 94.3% and 93.5% on the scheduled training subset, respectively. It achieves 93.8% and 92.9% on the signal processing subset, respectively. In contrast, the decrease of Model-B, Model-C and Model-D is more obvious in the modules with high timing dependence, especially in the two subsets of driving simulation and scheduling training, where the inter-model gap is further extended.

Table 6: Comparison of prediction performance on different teaching modules

Module	Metric	Model-A	Model-B	Model-C	Model-D
Theoretical Learning	Accuracy	94.0	91.6	89.7	88.1
Theoretical Learning	Macro-F1	93.2	90.8	88.5	87.0
Driving Simulation	Accuracy	95.1	92.0	89.4	87.6
Driving Simulation	Macro-F1	94.4	91.1	88.2	86.9
Dispatch Training	Accuracy	94.3	91.2	88.9	87.1
Dispatch Training	Macro-F1	93.5	90.4	87.8	86.2
Signal Handling	Accuracy	93.8	90.9	88.4	86.8
Signal Handling	Macro-F1	92.9	89.8	87.2	85.9

Table 6 shows that the proposed model does not only achieve local advantages on a certain type of data, but maintains strong classification ability in both knowledge learning

tasks and operation execution tasks. Especially in the two types of tasks of driving simulation and scheduling training, the advantages of Model-A over the Model are more obvious, indicating that the stability index, structure score and calibration output mechanism constructed in the previous section have A better adaptation effect on high procedural samples.

In summary, Model-A is superior to the comparison models in terms of training convergence speed, sample fitting consistency, test set classification performance, repeated trial stability, cross-module transfer ability and module synergy effect. The overall performance shows that the performance prediction model of rail transit teaching students constructed in this paper can not only maintain high accuracy and low error rate under the condition of full sample, but also maintain a stable output level in different teaching modules such as theoretical learning, driving simulation, scheduling training and signal disposal. Ablation experiments further show that behavioral weighted mapping, stability index, structure score and temperature calibration together constitute the main source of model performance improvement. Based on the above results, the proposed model can provide a reliable calculation basis for student hierarchical identification, training process tracking and teaching feedback generation in rail transit teaching.

5 Discussion

Give a strong advantage. The experimental results show that the Accuracy of Model-A on the test set reaches 94.6%, Macro-F1 reaches 93.8%, AUC is 0.962, ECE is only 0.028, the average accuracy of 10-fold cross validation is 93.9%, and the standard deviation is controlled within 0.47%. This shows that the model can not only distinguish excellent, standard and early warning samples more accurately, but also maintain high consistency under different data partition conditions. Compared with Model-B, Model-C and Model-D, Model-A has more obvious advantages in high procedural modules such as driving simulation, scheduling training and signal processing. The Accuracy of driving simulation subset reaches 95.1%, and Macro-F1 reaches 94.4%. It shows that the weighted behavior mapping, stability index and structure score are more expressive for complex training behaviors.

From the perspective of model mechanism, the performance improvement is not brought by a single classifier, but is formed by multi-source data collation, procedural feature modeling, gradient boosting training, and probabilistic calibration. Ablation experiments show that the Accuracy decreases from 94.6% to 91.8% after removing the behavior weighted mapping, and the AUC decreases only slightly after removing the temperature calibration, but the ECE increases from 0.028 to 0.067, indicating that the input expression and output calibration correspond to the key link of the model's discriminant ability and probabilistic consistency, respectively. In this paper, the identity information is anonymized and desensitized in the data processing stage, and only the fields directly related to learning behavior and training performance are retained in the training process, so as to ensure a clear boundary of data use. It should be noted that the current experimental samples are mainly from 5240 teaching records from 2019 to 2024, and the sample sources are still focused on limited scenarios. Subsequent research can continue to verify the adaptation ability of the model in a wider range of rail transit teaching tasks on the basis of expanding the sources of institutions, increasing the granularity of equipment logs and introducing cross-cycle tracking data. After preprocessing, Model-A enters the stable convergence interval around the 42nd round, reaching a similar training state 29 rounds earlier than the non-preprocessing condition. This shows that uniform encoding and temporal alignment not only improve the consistency of the input data, but also reduce the invalid consumption during the model iteration, thus

compressing the overall training cost. Based on this feature, the model is more suitable for student hierarchical identification, training process tracking and teaching feedback generation in rail transit teaching, and its application direction is more clear.

6 Conclusions

Focusing on the task of student performance prediction in rail transit teaching, this paper constructs a framework consisting of multi-source data collation, feature modeling, machine learning training, and calibration output. Experimental results show that the Accuracy of Model-A on the test set reaches 94.6%, Macro-F1 reaches 93.8%, AUC is 0.962, ECE is 0.028, 10-fold cross validation average accuracy is 93.9%, and standard deviation is 0.47%. In the driving simulation module, the Accuracy reaches 95.1% and Macro-F1 reaches 94.4%. These results show that the proposed model can stably identify three types of students: excellent, standard and early warning, and provide a computational basis for training stratification and feedback generation.

It should be noted that the current experimental samples mainly come from 5240 teaching records from 2019 to 2024, and the sample sources are still focused on limited scenarios. Although theoretical learning, driving simulation, scheduling training and signal disposal have been covered, the data differences under different colleges, different equipment platforms and different training systems have not been fully developed. At the same time, the existing labeling system is still based on the joint calculation of stage performance, process behavior and rule execution state, and there is still room for refinement in the description of long-term ability transfer, so the model conclusion is more suitable to explain the performance change in the current training cycle.

The follow-up research can be further promoted from three directions. First, the cross-college, cross-device and cross-cycle sample library should be expanded to enhance the migration and adaptation ability of the model in a wider range of tasks. Secondly, more fine-grained device log, operation trace and behavior sequence information are introduced to further improve the quality of feature representation. Thirdly, the mechanism of result interpretation, confidence output and boundary sample identification are strengthened while maintaining the classification performance, so that the model can not only give prediction results, but also support teaching adjustment, process tracking and continuous evaluation.

Funding

This work was supported by the National Natural Science Foundation of China (Grant No.2345678).

References

[1] Yağcı M. Educational data mining: prediction of students' academic performance using machine learning algorithms[J]. Smart Learning Environments, 2022, 9(1): 11.

[2] Chen H C, Prasetyo E, Tseng S S, et al. Week-wise student performance early prediction in virtual learning environment using a deep explainable artificial intelligence[J]. Applied Sciences, 2022, 12(4): 1885.

[3] Aljaloud A S, Uliyan D M, Alkhalil A, et al. A deep learning model to predict student

learning outcomes in LMS using CNN and LSTM[J]. IEEE Access, 2022, 10: 85255-85265.

[4] Jawad K, Shah M A, Tahir M. Students' academic performance and engagement prediction in a virtual learning environment using random forest with data balancing[J]. Sustainability, 2022, 14(22): 14795.

[5] Liu T, Wang C, Chang L, et al. Predicting high-risk students using learning behavior[J]. Mathematics, 2022, 10(14): 2483.

[6] Li M, Wang X, Wang Y, et al. Study-GNN: a novel pipeline for student performance prediction based on multi-topology graph neural networks[J]. Sustainability, 2022, 14(13): 7965.

[7] Kaur H, Kaur T, Garg R. A prediction model for online student academic performance using machine learning[J]. Informatica, 2023, 47(1).

[8] Alija S, Beqiri E, Gaafar A S, et al. Predicting students performance using supervised machine learning based on imbalanced dataset and wrapper feature selection[J]. Informatica, 2023, 47(1).

[9] Chen Y, Wei G, Liu J, et al. A prediction model of student performance based on self-attention mechanism[J]. Knowledge and Information Systems, 2023, 65(2): 733-758.

[10] Nayak P, Vaheed S, Gupta S, et al. Predicting students' academic performance by mining the educational data through machine learning-based classification model[J]. Education and Information Technologies, 2023, 28(11): 14611-14637.

[11] Liu Y, Huang Z, Wang G. Student learning performance prediction based on online behavior: an empirical study during the COVID-19 pandemic[J]. PeerJ Computer Science, 2023, 9: e1699.

[12] Wen X, Juan H. Early prediction of students' performance using a deep neural network based on online learning activity sequence[J]. Applied Sciences, 2023, 13(15): 8933.

[13] Xiao W, Hu J. A state-of-the-art survey of predicting students' performance using artificial neural networks[J]. Engineering Reports, 2023, 5(8): e12652.

[14] Lee J E, Jindal A, Patki S N, et al. A comparison of machine learning algorithms for predicting student performance in an online mathematics game[J]. Interactive Learning Environments, 2024, 32(9): 5302-5316.

[15] Fazil M, Rísquez A, Halpin C. A novel deep learning model for student performance prediction using engagement data[J]. Journal of Learning Analytics, 2024, 11(2): 23-41.

[16] Shou Z, Xie M, Mo J, et al. Predicting student performance in online learning: a multidimensional time-series data analysis approach[J]. Applied Sciences, 2024, 14(6): 2522.

[17] Ren Y, Yu X. Long-term student performance prediction using learning ability

self-adaptive algorithm[J]. Complex & Intelligent Systems, 2024, 10(5): 6379-6408.

[18] Luo Z, Mai J, Feng C, et al. A method for prediction and analysis of student performance that combines multi-dimensional features of time and space[J]. Mathematics, 2024, 12(22): 3597.

[19] Alnasyan B, Basheri M, Alassafi M. The power of Deep Learning techniques for predicting student performance in Virtual Learning Environments: A systematic literature review[J]. Computers and Education: Artificial Intelligence, 2024, 6: 100231.

[20] Hernández-García Á, Cuenca-Enrique C, Del-Río-Carazo L, et al. Exploring the relationship between LMS interactions and academic performance: a learning cycle approach[J]. Computers in Human Behavior, 2024, 155: 108183.