Predicting Student Grades via Adaptive Multi-Level Learning Models

Ali Reza Ibrahimzada[1], Kerem Kosif[2], Ahmed Said Gulsen[3], Yavuz Selim Yaglica[3] and Ali Cakmak[3]*
[1] University of Illinois Urbana-Champaign, Champaign, IL 61801
[2]Bogazici University, Bebek, Istanbul, Turkiye
[3] Istanbul Technical University, Maslak, Istanbul, Turkiye

*Corresponding author: ali.cakmak@itu.edu.tr

Abstract

Educational institutions increasingly rely on intelligent systems to extract actionable insights from student data. One critical application is the early prediction of student performance in specific courses, which can inform academic advising, course selection, and targeted interventions. This paper proposes an adaptive multi-level prediction framework that segments student-course data into homogeneous groups and assigns a temporally validated specialist model to each group. The framework is model-agnostic: it accepts any learner implementing standard `fit`/`predict` interfaces, including linear regression, ensemble methods, neural networks, and collaborative filtering. To combat temporal data sparsity common in volatile or block-cohort curriculum structures, the framework incorporates an automated data-density fallback guard that dynamically transitions isolated data slices to localized validation pools.

Systematic validation on two large-scale, real-world higher education datasets demonstrates strong cross-institutional generalizability. On a dense traditional university dataset, the framework yields an 18.6% RMSE improvement over global baselines, with model selection converging heavily on tree-based ensembles. Conversely, on a volatile sparse course dataset, the pipeline automatically uncovers an extraordinarily diverse model ecosystem—selecting neural layers for 48% of clusters and triggering the collaborative filtering 22% of chronological windows. Backed by asymptotic significance testing (p-val $< 10^{-207}$), these results prove that the framework effectively shifts the configuration burden from manual heuristics to self-correcting, data-driven optimization.

Keywords: Data science applications in education, Adaptive Regression, Educational Data Mining, Academic Performance Forecasting, Cluster-Based Specialization

1. Introduction

The rapid adoption of information systems in educational institutions has increased the amount of collected data, which plays a critical role in decision-making processes. The ability to predict future course grades can be invaluable for students, instructors, and administrators. Students can use intelligent agents to make informed decisions about elective courses, avoid extra expenses, and graduate on time. Instructors benefit from early predictions by tailoring course materials and identifying at-risk students for timely intervention. At the institutional level, administrators can leverage predictive insights to monitor instructional effectiveness and optimize resource allocation.

Several works have studied the problem of student success prediction [1, 2, 3, 4, 5]. [6] provide a comprehensive survey of educational data mining research, highlighting two key dimensions: (i) the timing of prediction and (ii) the nature of the predicted outcome. Most existing studies focus on predicting categorical success outcomes (e.g., pass/fail) during the course, largely due to the availability of rich in-course data. However, two critical challenges remain underexplored: (i) predicting exact letter grades rather than broad categories, and (ii) making these predictions *before* the course begins.

While some studies have addressed the grade prediction problem, many approaches rely on student participation in data collection (e.g., surveys, course management system logs, SAT scores), which limits widespread adoption. Others are constrained by small datasets, reducing generalizability. As one of the most recent relevant studies, [7] proposed a knowledge graph-enhanced collaborative filtering model integrating course descriptions and external content. While effective, this method requires extensive data collection, making it more suitable for MOOC platforms. In contrast, our method is designed for traditional academic settings and requires only historical grade data.

To address these limitations, we propose an adaptive prediction framework that segments student-course data into homogeneous groups and assigns a tailored predictive model to each group, selected via temporal-validation from a pool of nine candidates. This modular architecture eliminates the one-size-fits-all assumption, reduces variance, and is generalizable across institutions. We validate the framework on two real-world higher education datasets containing approximately 55,000 and 19,000 course-grade records, respectively, and show that the partition-and-specialize strategy consistently improves accuracy over both baseline and state-of-the-art methods.

Contributions. Our main contributions are as follows:

- We introduce an adaptive prediction framework that segments student-course data and assigns a temporally-validated specialist model to each segment, improving prediction accuracy and model fit.
- The framework is *model-agnostic*: it defines a Generic Predictor Interface requiring only standard $\texttt{fit}(X, y)$ and $\texttt{predict}(X)$ methods, enabling seamless integration of any statistical learning approach without modifying the core partitioning logic.

- We provide the first systematic per-cluster model selection experiment, evaluating eight regression algorithms and collaborative filtering as competing candidates via temporal-validation. Regression wins every cluster in every semester, achieving an 18.6% RMSE improvement over the global collaborative filtering baseline on identical test instances.
- We systematically evaluate two segmentation strategies—student-based and course-based—and quantify their impact on predictive performance and cluster quality (Silhouette analysis).
- We perform extensive feature engineering and propose a rich set of predictors derived entirely from existing institutional records, without requiring surveys or external data.
- We introduce a native data-density guard capable of preserving chronological execution continuity when exposed to severe curriculum volatility or block-cohort student dynamics. We show that our self-correcting framework dynamically senses localized data starvation and deploys resilient fallback routing without altering global evaluation baselines.
- Through rigorous external validation on fundamentally contrasting educational data structures, we demonstrate that optimal predictor selection is inherently dataset-driven. We provide empirical proof that dense, continuous university registries naturally optimize under variance-reducing tree ensembling architectures, whereas sparse, highly volatile datasets require complex non-linear deep learning models and collaborative matrix structures to sustain predictive stability.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 presents the methodology: the framework architecture and algorithm (Sections 3.1–3.3), parameter selection guidance (Section 3.4), a catalogue of candidate models (Section 3.5), and reproducibility details (Section 3.6). Section 4 reports experimental results. Section 5 discusses broader implications and limitations. Section 6 concludes the paper.

2. Related Work

Educational Data Mining (EDM) has emerged as a prominent field focused on extracting actionable insights from academic data [6, 8, 9]. A wide range of methodologies have been proposed for predicting student performance, which can be broadly categorized into five major groups: clustering-based models, decision-theoretic approaches, matrix factorization and hybrid models, machine learning-based predictors, and knowledge graph-enhanced systems.

2.1. Clustering-Based Models

[5] proposed a probabilistic clustering model using student questionnaires to predict final grades. While conceptually similar to ours, it relies on manually

collected data and lacks scalability. [10] also explored clustering by grouping students based on grade patterns, but did not integrate predictive modeling within clusters. Our work extends this by combining clustering with per-cluster temporally-validated model selection.

2.2. Decision-Theoretic Approaches

[11] employed a Markov Decision Process (MDP) to model student course selection and performance. While MDPs offer flexibility, they require extensive feature engineering. Our method avoids these complexities by using K-Means clustering and a common `fit`/`predict` interface for all candidate models.

2.3. Matrix Factorization and Hybrid Models

Matrix factorization (MF) and hybrid models have been widely used for grade prediction. [12] proposed a Factorization Machine–Random Forest hybrid. [13] extended MF with SAT scores and course syllabi. [14] developed course-specific regression models and demonstrated their superiority over student-based collaborative filtering. However, these methods often depend on data unavailable in many institutions. Our framework avoids this limitation by relying solely on course enrollment and grade history.

2.4. Machine Learning-Based Predictors

Numerous studies have applied machine learning techniques to student performance prediction, including neural networks [15, 16, 17, 18, 19], SVMs [20, 21, 22, 23, 24], ensemble methods [25, 26, 27, 28, 29], and Naive Bayes classifiers [30, 31, 32, 33, 34]. Many of these models incorporate demographic or behavioral data that may not be accessible due to privacy concerns. Our approach circumvents this by using only academic records. Several works also address class imbalance using SMOTE [15, 25, 35, 16, 36, 17, 26, 18, 27, 28, 29, 37, 24], but do not address the heterogeneity of student populations; our clustering-based framework explicitly tackles this.

2.5. Cross-Domain Advances in Predictive Modeling

Liu et al. [38] introduced a detracking autoencoding conditional GAN for tabular missing-value imputation, demonstrating robust strategies for sparse feature matrices— relevant to our fallback design. Moreover, Liu et al. [39] proposed hybrid neural network innovations for time-series forecasting whose core contributions in modeling temporal dependencies align with our temporal validation framework. Our work adapts these rigorous predictive standards to the educational domain.

2.6. Summary

Prior work has explored collaborative filtering, regression, and hybrid approaches, each with specific advantages and drawbacks. A common limitation is the application of a single predictive model uniformly across all student and course contexts. Our framework addresses this by enabling *data-driven model specialization* per cluster, using only readily available institutional data.

3. Methodology

This section presents the full methodology. We begin by formally defining the prediction problem (Section 3.1), then describe the adaptive framework and its algorithmic components (Sections 3.2–3.2.6). Section 3.3 provides parameter selection guidance. Section 3.4 catalogues the nine model candidates. Section 3.5 documents reproducibility details.

3.1. Problem Formulation

Let $\mathcal{S}$ denote the set of students, $\mathcal{C}$ the set of courses, and $\mathcal{T} = \{t_1, t_2, \ldots, t_T\}$ the ordered sequence of academic semesters. Each student–course–semester triple (s, c, t) for which a letter grade $g \in \mathcal{G}$ is recorded constitutes one instance. We map each letter grade to a numerical value (see Table 3) and treat the problem as regression.

Given all observed grades up to semester t_n, the goal is to predict the grade $\hat{g}_{s,c}$ that student s will receive in course c during semester t_{n+1}, using no information from t_{n+1} or later. This strict temporal constraint is enforced throughout: the training partition S_{train} consists exclusively of records from semesters $\{t_1, \ldots, t_n\}$, and all aggregate features are computed dynamically over S_{train} only (see Section 3.2.5).

3.2. Adaptive Multi-Level Prediction Framework

A single predictive model may fail to capture the diversity of patterns across different student and course profiles. We propose a modular prediction framework that partitions the training data into related subsets and fits a temporally-validated specialist model to each subset, enhancing predictive performance and interpretability.

Figure 1 illustrates the three main stages. First, all student-course instances are preprocessed: records with ungraded outcomes (e.g., Pass/Fail, Incomplete) are removed, and the remaining data is split according to the temporal constraint above. Second, the training instances are partitioned into k groups via K-Means clustering on student or course features (Section 3.2.1). Third, for each cluster a dedicated model is selected from the candidate pool $\mathcal{C}$ based on performance evaluation via a complementary seasonal validation split. Instead of a random hold-out, validation is performed on the historical semester that mathematically matches the academic season of the target semester (e.g., validating on the previous Spring to predict the current Spring), while all other historical data is utilized for training. The selected model per cluster is retrained on the full cluster data. During testing, we do not hard-cluster a test instance t to the closest cluster and use that cluster's model to predict its grade. Instead, we soft-cluster t to multiple clusters, and employ a weighted ensemble of the corresponding cluster models to predict the grade for t.

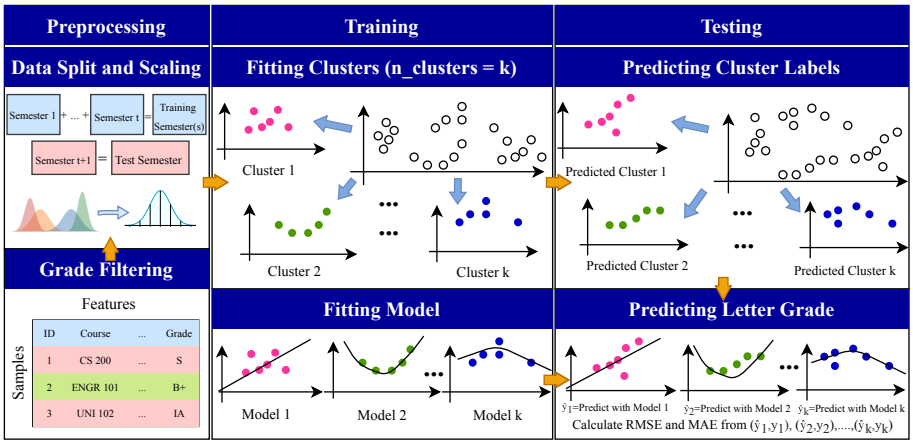

Figure 1: Main stages of the proposed adaptive framework: preprocessing and temporal split, unsupervised partitioning into k clusters, per-cluster temporally-validated model selection and training, and adaptive inference on the test semester.

3.2.1. Adaptive Clustering Strategy

The number of clusters per temporal train/test semester split is determined adaptively based on the size and internal coherence of the training semesters. More specifically, for each training data split, a local sweep of k-values is performed by increasing k one by one until a previously seen cluster structure appears again. This point indicates that increasing the value of k does not lead to new clusters. For each value of k, we run k-means clustering. In the resulting clustering, tiny clusters whose size is less than τ are merged with the closest cluster, starting from the smallest cluster. This way, very small clusters for which regression models cannot be trained properly are eliminated early on in the process. We discuss how τ is set in Section 3.3. We observed that as silhouette scores of clusterings decrease, the accuracies of the corresponding prediction models decreases. Motivated by this observation, we select the best 3 clustering configurations with the highest silhouette scores and train prediction models for them. Then, the k-value that provides the best training data error rate is chosen as the clustering configuration for this particular target semester. The maximum k value that we attempt is determined as $\lfloor training\_data\_size/\tau \rfloor$, as this is theoretically the highest number of clusters that k-means can achieve with a minimum cluster size of τ. Algorithm 1 describes the adaptive clustering strategy in a more concise way.

Algorithm 1: Adaptive Clustering Strategy

Input : $\mathcal{D}_{\text{train}}$: training data; τ : minimum cluster size
Output: $\mathcal{T}$: Top-3 clustering configurations with highest silhouette scores

Function *AdaptiveClusteringStrategy*($\mathcal{D}_{train}, \tau$)**:**
  `// Phase 1: Determine Search Range for k`
  $n \leftarrow |\mathcal{D}_{\text{train}}|$
  $k_{\max} \leftarrow \lfloor n/\tau \rfloor$; $k_{\min} \leftarrow 2$
  `// Phase 2: Sweep k Values and Detect Recurrence`
  $\mathcal{S}_{\text{previous}} \leftarrow \emptyset$; $\mathcal{K}_{\text{valid}} \leftarrow \emptyset$
  for $k \leftarrow k_{\min}$ **to** $k_{\max}$ **do**
    $\mathcal{C}_k \leftarrow \texttt{KMeans}(\mathcal{D}_{\text{train}}, k)$
    `// Phase 3: Merge Undersized Clusters`
    $\mathcal{C}_k^{\text{merged}} \leftarrow \texttt{MergeSmallClusters}(\mathcal{C}_k, \tau)$ `// See Alg. 2`
    $\mathcal{S}_k \leftarrow \texttt{GetClusterStructure}(\mathcal{C}_k^{\text{merged}})$
    if $\mathcal{S}_k \in \mathcal{S}_{previous}$ **then break** `// stop sweep if repeated`
    $\mathcal{S}_{\text{previous}} \leftarrow \mathcal{S}_{\text{previous}} \cup \{\mathcal{S}_k\}$
    $\mathcal{K}_{\text{valid}} \leftarrow \mathcal{K}_{\text{valid}} \cup \{(\mathcal{C}_k^{\text{merged}}, k)\}$
  end
  `// Phases 4 & 5: Silhouette Ranking & Top Selection`
  $\mathcal{R} \leftarrow \emptyset$
  foreach $(\mathcal{C}_k^{merged}, k) \in \mathcal{K}_{valid}$ **do**
    $\sigma_k \leftarrow \texttt{SilhouetteScore}(\mathcal{D}_{\text{train}}, \mathcal{C}_k^{\text{merged}})$
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{(k, \mathcal{C}_k^{\text{merged}}, \sigma_k)\}$
  end
  $\mathcal{R} \leftarrow \texttt{Sort}(\mathcal{R}, \text{by } \sigma_k \text{ descending})$
  $\mathcal{T} \leftarrow \text{Top-3}(\mathcal{R})$ `// Return the best 3 candidates`
  return $\mathcal{T}$

Algorithm Parameters and Implementation Notes

- **Threshold** τ: Minimum number of samples required per cluster. Recommended: $\tau \geq 10 \times D$, where D is the number of features (Section 3.3).
- **Cluster Structure** $\mathcal{S}_k$: A canonical representation of cluster membership (e.g., sorted list of cluster sizes, or hash of centroid positions). Used to detect when increasing k no longer produces new partitions.
- **Silhouette Score**: Measures cluster cohesion (how similar points are to their own cluster) and separation (how dissimilar points are from other clusters). Used to rank configurations: higher silhouette indicates better-defined clusters.
- **Top-3 Selection**: Selects the three clustering configurations with the highest silhouette scores.

Algorithm 2: Subroutine: Merge Undersized Clusters

Input : $\mathcal{C}$: initial clustering with centroids
τ : minimum cluster size threshold
Output: $\mathcal{C}^{\text{merged}}$: clustering after merging undersized clusters

Function *MergeSmallClusters*($\mathcal{C}, \tau$)**:**
  merged $\leftarrow$ False
  while *not* *merged* **do**
    sizes $\leftarrow \{|c| : c \in \mathcal{C}\}$ `// cluster sizes`
    if $\min(\textit{sizes}) \geq \tau$ **then**
      merged $\leftarrow$ True `// all clusters meet minimum`
    else
      $c_{\text{small}} \leftarrow \arg\min_c |c|$ `// smallest cluster`
      $c_{\text{near}} \leftarrow \arg\min_{c' \neq c_{\text{small}}} \|\mathbf{v}_{c_{\text{small}}} - \mathbf{v}_{c'}\|_2$ `// nearest centroid`
      $c_{\text{near}} \leftarrow c_{\text{near}} \cup c_{\text{small}}$ `// merge clusters`
      $\mathcal{C} \leftarrow \mathcal{C} \setminus \{c_{\text{small}}\}$
      Recompute centroid of c_{near}
    end
  end
  return $\mathcal{C}$

Clustering Features

We investigate two partitioning strategies:

1. *Student-based Partitioning:* Student-course instances are grouped on student-level attributes—GPA, completed credits, and department. The goal is to identify groups of students with similar academic profiles so that a specialist model can capture sub-population-specific grade distributions.

2. *Course-based Partitioning:* Instances are grouped on course-level characteristics—grade distribution, mean and standard deviation of previous students' grades in that course. This enables modeling of course-specific performance patterns.

The full feature sets for both strategies are defined in Table 1.

3.2.2. Generic Predictor Interface and Per-Cluster Model Selection

The framework defines a **Generic Predictor Interface** requiring only standard `fit`(X, y) and `predict`(X) methods. Any supervised learner that exposes these methods, i.e., regression, ensemble, neural network, or collaborative filtering, can be plugged into any cluster without modifying the partitioning logic. The candidate set $\mathcal{C}$ used in this study contains nine models (Section 3.4).

For each cluster j, the best model is selected based on performance evaluation via a timeline-based seasonal validation split. Because student grade distributions frequently exhibit distinct seasonal dynamics (i.e., Fall versus Spring course offerings and student loads), we match the validation and training data

Table 1: Feature sets for the unsupervised clustering phase.

Strategy	Feature	Description
Student-based	Cumulative GPA	Credit-weighted GPA over training semesters.
	Completed Credits	Total credits passed in training semesters.
	Department Code	One-hot encoded enrolled department.
Course-based	Mean Grade	Mean grade in the course (training data).
	STDEV Grade	Std. dev. of grades in the course (training data).
	Grade Distribution	Percentage of each letter grade in the course (training data).

distributions to the academic season of the target prediction semester. For example, to predict a Fall semester, we extract the most recent Fall semester from the historical training data to act as the validation set. The remaining historical Fall semesters are used to train the candidate models for selection. A parallel logic applies when predicting a Spring semester.

To handle early chronological forecasting where historical data is severely limited, we apply two fallback conditions:

1. **Single Historical Semester:** If the training data contains only one semester total, we apply a random 80/20 train-validation split on that single semester to evaluate candidates.

2. **Two Historical Semesters:** If the training data contains exactly two historical semesters (e.g., predicting Semester 3 [Fall] using Semester 1 [Fall] and Semester 2 [Spring]), the most recent Fall semester (Semester 1) acts as the validation set. Because no prior Fall semesters remain, the candidate models are trained on the remaining Spring semester (Semester 2).

Handling Distorted Slices: To ensure cross-institutional resilience, the framework incorporates an automated data-density guard during the parameter tuning phase. If highly localized clustering splits an external dataset into an archetype that yields zero historical training records under strict chronological sequencing (resulting in an uncomputable or empty training matrix slice), the pipeline triggers an automated fallback mechanism. It transitions that localized slice to a randomized cross-validation paradigm (an 80/20 split) to execute the model sweep and select the winner for that isolated segment. This safeguards the adaptive model selection loop against data-scarcity anomalies without altering the overarching chronological boundary conditions of the outer evaluation framework.

Once the optimal model $\mathcal{M}_j^*$ is selected for cluster j by minimizing the validation error:

$$\mathcal{M}_j^* = \arg\min_{\mathcal{M}\in\mathcal{C}} \overline{\mathrm{RMSE}}\big(\mathcal{M},\, X_{train\_candidates}[C_j],\, Y_{val\_season}[C_j]\big). \quad (1)$$

The winning model is then finally retrained exclusively on *all* historical semesters matching the target season (e.g., all Fall semesters) to maximize seasonal data utilization while preventing cross-season noise.

3.2.3. Predictor Attributes (Features) for Regression

- *Course Level:* Derived from the first digit of the course code. Codes 1–4 denote Undergraduate; 5 Graduate; 6 PhD.
- *Course Year in Curriculum:* First digit of the course code (e.g., CS 201 $\rightarrow$ Year 2).
- *GPA:* Credit-weighted grade point average computed exclusively over training-partition courses, preventing data leakage.
- *Completed Credits:* Sum of credit hours of successfully passed courses in the training partition only.
- *Average GPA – Subject:* Mean GPA of students in courses sharing the same subject prefix (training data only; see Section 3.2.5).
- *Average Grade – Course:* Mean grade received by prior students in the same course (training data only).
- *Letter Grade Percentages:* Percentage of each letter grade (A+ through F) received in the course in the training data (13 binary subfeatures).

3.2.4. Feature Preprocessing

One-Hot Encoding. Categorical variables—*Department Code*, *Course Level*, and *Course Year*—lack an inherent ordinal relationship suited for Euclidean-distance-based clustering. We apply one-hot encoding:

$$v_i = \begin{cases} 1 & \text{if the sample belongs to category } i \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

This ensures all categories are treated equidistantly.

Z-Score Normalization. Numerical features span very different ranges (e.g., GPA $\in [0, 4.1]$ vs. Completed Credits $\in [0, 240+]$). To prevent high-magnitude features from dominating K-Means, we apply StandardScaler to all numerical features before clustering and regression:

$$x' = \frac{x - \mu}{\sigma} \quad (3)$$

where μ and σ are computed on the training set and then applied to test features.

3.2.5. Prevention of Data Leakage in Feature Engineering

All aggregate features (*Average GPA – Subject* and *Average Grade – Course*) are computed dynamically and exclusively over the training partition S_{train}. Formally, the *Average GPA – Subject* for subject j is:

$$\mu_{\text{subj}_j} = \frac{\sum_{(s,c,g)\in S_{train}} g \cdot \mathbb{I}(c \in \text{subj}_j)}{\sum_{(s,c,g)\in S_{train}} \mathbb{I}(c \in \text{subj}_j)} \tag{4}$$

where $\mathbb{I}$ is the indicator function. Records from the test semester are strictly excluded from S_{train}, and these averages are recomputed from scratch at every temporal split, ensuring realistic forecasting constraints.

3.2.6. Course Grade Prediction Algorithm

Algorithm 3 formalizes the full prediction workflow in three phases. In temporal order, we iterate over semesters starting with the second semester. For each target semester, we set the previous semesters' courses as training data, and the target semester's courses as target data. In the first phase, we perform adaptive clustering for each target semester (Algorithm 1).

In the second phase, for each cluster, we select the best model via a timeline-based seasonal validation among all learning models included in set $\mathcal{C}$. We designate the most recent historical semester of the target season (Fall or Spring) as the validation set, and train candidates on the remaining historical semesters of that same season. (Fallback strategies, such as an 80/20 split or cross-season training, are applied for the earliest semesters where historical data is insufficient). Finally, the winning model is retrained on all historical data belonging exclusively to the target season.

In the third phase, each test instance c is soft-assigned to multiple clusters, where each membership with a cluster i is weighted by a Euclidean distance-based similarity (i.e., $1/Euclidean\_Distance(centroid_i, c)$). The weights are normalized so that they sum up to 1. Grade for a course c is predicted by combining the predictions from multiple cluster models weighted by the similarity between c and each cluster.

Algorithm 3: Adaptive Multi-Level Prediction Framework

Input : $\mathcal{S} = \{S_1, S_2, \ldots, S_T\}$: Sequence of chronological semesters
$\mathcal{C}$: Set of prediction models (e.g., RF, LR, MLP, SVR, CF)
τ : Minimum cluster size threshold

Output: $\hat{Y}$: Predicted grades across all target semesters

$\hat{Y} \leftarrow \emptyset$
for $t \leftarrow 2$ **to** T **do**
  $\mathcal{D}_{\text{train}} \leftarrow \bigcup_{i=1}^{t-1} S_i$; $\mathcal{D}_{\text{test}} \leftarrow S_t$ `// Define temporal split`
  `// Phase 1: Unsupervised Partitioning Candidates`
  $\mathcal{T} \leftarrow$ `AdaptiveClusteringStrategy`$(\mathcal{D}_{\text{train}}, \tau)$ `// See Alg. 1`
  `// Phase 2: Configuration & Model Selection`
  $E_{\text{best}} \leftarrow \infty$; $\mathcal{H}_{\text{best}} \leftarrow \emptyset$; $\mathbf{V}^* \leftarrow \emptyset$; $k^* \leftarrow 0$
  foreach *configuration* $(k, \mathcal{C}_k^{merged}, \sigma_k) \in \mathcal{T}$ **do**
    $E_{\text{config}} \leftarrow 0$; $\mathcal{H}_{\text{temp}} \leftarrow \emptyset$
    foreach *cluster* $j \in \{1, \ldots, k\}$ **do**
      $\mathcal{D}_{\text{train}}^{(j)} \leftarrow \{x \in \mathcal{D}_{\text{train}} \mid \text{assign}(x) = j\}$
      $\mathcal{D}_{val}^{(j)} \leftarrow$ Most recent semester in $\mathcal{D}_{\text{train}}^{(j)}$ matching season of S_t
      $\mathcal{D}_{train\_sub}^{(j)} \leftarrow$ Remaining sem. in $\mathcal{D}_{\text{train}}^{(j)}$ matching season of S_t
      if $|\mathcal{D}_{train\_sub}^{(j)}| == 0$ ***or*** $\mathcal{D}_{val}^{(j)} == \emptyset$ **then**
        $\mathcal{D}_{train\_sub}^{(j)}, \mathcal{D}_{val}^{(j)} \leftarrow$ `RandomSplit`$(\mathcal{D}_{\text{train}}^{(j)}, \text{ratio} = 0.80)$
      end
      $m_j^* \leftarrow \arg\min_{m \in \mathcal{C}} \text{ValidationError}(m, \mathcal{D}_{train\_sub}^{(j)}, \mathcal{D}_{val}^{(j)})$
      if $m_j^* == None$ **then**
        $\mathcal{D}_{train\_sub}^{(j)}, \mathcal{D}_{val}^{(j)} \leftarrow$ `RandomSplit`$(\mathcal{D}_{\text{train}}^{(j)}, \text{ratio} = 0.80)$
        $m_j^* \leftarrow \arg\min_{m \in \mathcal{C}} \text{ValidationError}(m, \mathcal{D}_{train\_sub}^{(j)}, \mathcal{D}_{val}^{(j)})$
      end
      $E_{\text{config}} \leftarrow E_{\text{config}} + \text{ValidationError}(m_j^*)$ `// Accum. error`
      $\mathcal{D}_{season}^{(j)} \leftarrow \mathcal{D}_{val}^{(j)} \cup \mathcal{D}_{train\_sub}^{(j)}$; $\mathcal{H}_{\text{temp}}[j] \leftarrow m_j^*$.`fit`$(\mathcal{D}_{season}^{(j)})$
    end
    if $E_{config} < E_{best}$ **then**
      $E_{\text{best}} \leftarrow E_{\text{config}}$; $\mathcal{H}_{\text{best}} \leftarrow \mathcal{H}_{\text{temp}}$
      $k^* \leftarrow k$; $\mathbf{V}^* \leftarrow$ centroids of $\mathcal{C}_k^{\text{merged}}$
    end
  end
  `// Phase 3: Adaptive Soft-Ensemble Inference`
  foreach *test instance* $x \in \mathcal{D}_{test}$ **do**
    $W_{\text{total}} \leftarrow 0$; $\hat{y}_x \leftarrow 0$
    foreach *cluster* $j \in \{1, \ldots, k^*\}$ **do**
      $d_j \leftarrow$ `Euclidean_Distance`$(x, \mathbf{v}_j^*)$
      $w_j \leftarrow 1/(d_j + \epsilon)$ `// ϵ avoids division by zero`
      $W_{\text{total}} \leftarrow W_{\text{total}} + w_j$
    end
    foreach *cluster* $j \in \{1, \ldots, k^*\}$ **do**
      $\tilde{w}_j \leftarrow w_j / W_{\text{total}}$
      $\hat{y}_x \leftarrow \hat{y}_x + \tilde{w}_j \cdot \mathcal{H}[j]$.`predict`$(x)$
    end
    $\hat{Y} \leftarrow \hat{Y} \cup \{(x, \hat{y}_x)\}$
  end
end
return $\hat{Y}$

3.3. Determining the Minimum Cluster Size Threshold (τ)

The sparsity threshold τ prevents fitting models on clusters too small for reliable parameter estimation. Following the "one in ten" rule of regression [40], we recommend:

$$\tau \geq 10 \times D \tag{5}$$

where D is the number of input features. That is, in our algorithm $D = 59$ (after one-hot encoding), which yields $\tau = 590$ samples.

3.4. Employed Prediction Models

The Generic Predictor Interface accepts any supervised learner. This section lists the nine candidates in $\mathcal{C}$ evaluated in this study, organized by learning paradigm.

3.4.1. Linear Regression Models

Three linear models establish interpretable baseline performance within each cluster: *Ridge Regression* (L2 penalty, reduces multicollinearity), *Lasso Regression* (L1 penalty, implicit feature selection), and *Support Vector Regression* (SVR) with an RBF kernel (non-linear margin-based regression). These low-variance models are particularly effective for small or sparse clusters.

3.4.2. Ensemble Methods

Three ensemble methods capture non-linear patterns: *Random Forest* (bagging of decision trees, robust to outliers), *Gradient Boosting* (sequential residual correction), and *Bagging Regressor* (variance reduction via bootstrap aggregation). Ensemble methods are typically selected by CV for larger, data-rich clusters.

3.4.3. Neural Architecture

A *Multi-Layer Perceptron* (MLP) with two hidden layers (100, 50 neurons), ReLU activation, and Adam optimizer represents neural learning approaches. MLP can capture complex non-linear interactions but requires sufficient cluster data for stable gradient estimates.

3.4.4. Instance-Based Learning

The *k-Nearest Neighbors (kNN)* regressor predicts the target grade by interpolating the grades of the closest k instances in the feature space, weighted by their inverse distance. This localized, non-parametric approach is particularly robust when students with similar academic profiles consistently achieve similar outcomes, directly complementing the localized nature of our clustering framework.

3.4.5. Collaborative Filtering

Collaborative filtering (CF) [41] identifies latent similarities between students (users) or courses (items) from historical grades. We implement the item-based CF for efficiency with Euclidean Distance [42] as the similarity metric. To address CF's cold start case situations, where CF cannot create a prediction for new courses or students, student GPA (computed over training courses) is used as the prediction.

3.5. Reproducibility

All predictive models are implemented using scikit-learn. Hyperparameters are selected via grid search; final configurations are reported in Table 2. Besides, a fixed random_state=42 value is set whenever an underlying learning algorithm makes random value-based assignment/initialization.

4. Results

We quantitatively analyze the proposed techniques from various perspectives. Datasets and metrics are discussed first, followed by individual experimental findings.

4.1. Datasets

For experimental assessment, we use two datasets. The first is a student course grade dataset acquired from Istanbul Sehir University (SEHIR dataset). All student-specific information (student ID, name, nationality, etc.) was removed before provision.

The original SEHIR dataset spans 2010–2015 and contains 55,475 course grade instances. We removed 6,128 records: 3,891 pass/fail records and 2,237 incomplete grade records (11.0% of the total). The removed records are distributed uniformly across semesters and departments, with no single stratum exceeding 14% of its own records, confirming that the filtering does not introduce systematic selection bias. The final SEHIR dataset contains 49,347 instances from 2,759 distinct students. Figure 2 shows the grade distribution.

The second dataset is from the School of Big Data and Artificial Intelligence, Software Engineering program, Anhui Xinhua University [7] (ANHUI dataset). It contains normalized grades (1–5) for 382 students across 50 courses over 7 semesters (19,100 instances).

4.1.1. Grade Normalization and Encoding

Categorical letter grades in the SEHIR dataset were mapped to numerical values per Table 3. The ANHUI dataset uses a native 1–5 scale, retained without modification. Due to differing scales (0–4.1 vs. 1–5), RMSE and MAE values are *not* comparable across datasets; all comparisons are conducted within each dataset.

Table 2: Hyperparameter configuration for models.

Model	**Hyperparameters**
BaggingRegressor	n_estimators = 1000
	max_samples = 0.8
GradientBoostingRegressor	learning_rate = 0.05
	max_depth = 4
	loss = huber
	l2_regularization=0.5
	max_iter=1000
	n_iter_no_change = 15
	validation_fraction=0.1
KNNRegressor	n_neighbors = 5
	weights = distance
Lasso	$\alpha = 1.0$
RandomForestRegressor	n_estimators = 400
	max_depth = 12
	min_samples_leaf = 8
	max_features = sqrt
Ridge	$\alpha = 1.000$
SVR	kernel = rbf
	$C = 10$
	$\epsilon = 0.1$
	$\gamma = scale$
MLP Regressor	Hidden Layers = (128, 64, 32)
	Activation = relu
	Solver = adam
	$\alpha = 0.01$, max_iter = 1000

4.2. Metrics

We employ RMSE [43] (Equation 6) and MAE (Equation 7) as accuracy metrics.

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2} \tag{6}$$

$$\text{MAE}(y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}|y_i - \hat{y}_i| \tag{7}$$

To combine RMSE and MAE into a single score, inspired by F1 score for classification tasks, we employ Harmonic Error Metric (HEM) by computing

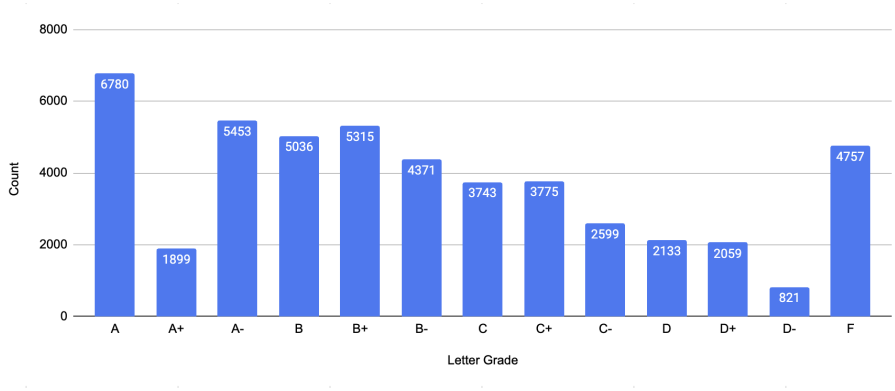

Figure 2: Grade distribution in the SEHIR dataset.

harmonic mean of RMSE and MAE (Eq. 8). Because the harmonic mean is strongly influenced by the lower value (MAE) , it naturally pushes the framework to select models with tight absolute precision. However, if a model has a catastrophic failure that causes its RMSE to spike, the denominator shifts and drastically punishes the HEM score. It forces the framework to find the best model that balances bias and variance. During model selection, we set the loss function as the HEM score.

$$HEM = \frac{2 \times RMSE \times MAE}{RMSE + MAE} \tag{8}$$

4.3. Experiments

All experiments use a strict temporal split as the evaluation protocol: training consists of all prior semesters $\{t_1, \ldots, t_n\}$ and testing is the immediately following semester t_{n+1}. This ensures zero temporal leakage and reflects realistic deployment conditions where future grades are unavailable at prediction time. All experiments were carried out on a DELL R720 server (24-core vCPU, 80 GB RAM, 2.4 TB storage) using Python 3.6.9 and scikit-learn.

4.4. Baseline Evaluation: Global Prediction Models ($k = 1$)

To establish a rigorous baseline and quantify the necessity of localized clustering, we first evaluate the performance of all candidate algorithms trained globally on the entire student population ($k = 1$). In this configuration, the temporal model selection mechanism is still applied, but it selects a single, global specialist model for the entire target semester rather than per-cluster.

4.4.1. Overall Global Performance

Figures 3, **??**, and **??** present the overall RMSE, MAE, and HEM across all test semesters for the standalone candidates and the global adaptive ensemble. Several key observations emerge from the global baseline evaluation:

Table 3: Letter grade to numerical value mapping (SEHIR dataset).

Letter Grade	Numeric Value	Description
A+	4.1	Excellent
A	4.0	Excellent
A-	3.7	Excellent
B+	3.3	Good
B	3.0	Good
B-	2.7	Good
C+	2.3	Average
C	2.0	Average
C-	1.7	Average
D+	1.3	Below Average
D	1.0	Poor
D-	0.5	Poor
F	0.0	Fail

- **Dominance of Ensemble Regression:** Tree-based ensemble methods achieve the best overall global performance. The Gradient Boosting Machine (GBM) leads the standalone candidates (RMSE 0.60, MAE 0.42, HEM 0.50), followed closely by the Bagging Regressor (RMSE 0.61, MAE 0.43). Their ability to handle non-linear interactions within the massive, unsegmented dataset gives them a distinct advantage over standard linear approaches like Ridge (RMSE 0.65) and Lasso (RMSE 0.82).

- **High-Fidelity Temporal Selection:** The global Adaptive Ensemble ($k = 1$) perfectly matches the performance of the best standalone algorithm (GBM) across all metrics (RMSE 0.60, MAE 0.42, HEM 0.50). This exact alignment is a critical validation of our chronological framework. It demonstrates that the target-season temporal validation mechanism successfully and consistently identifies the optimal predictive model using *only* historical data, achieving successful selection without leaking any future test-set patterns.

4.4.2. Temporal Trajectories and the Global Plateau

To analyze how model efficacy evolves as historical training data accumulates, Figure **??** traces the temporal error trajectories over the sequential test semesters (MAE and HAM figures are similar; hence, not included here).

The trajectory reveals a counter-intuitive but critical limitation of global modeling. In the earlier prediction windows (Semesters 1 through 3), where the test sets are relatively small (ranging from roughly 2,000 to 3,800 instances), the top-tier models successfully capture the dataset's variance, with the Adaptive Ensemble's RMSE dropping as low as 0.56.

However, as the chronological timeline progresses into Semesters 4 through 7, the volume and diversity of the academic data expand massively (surpassing

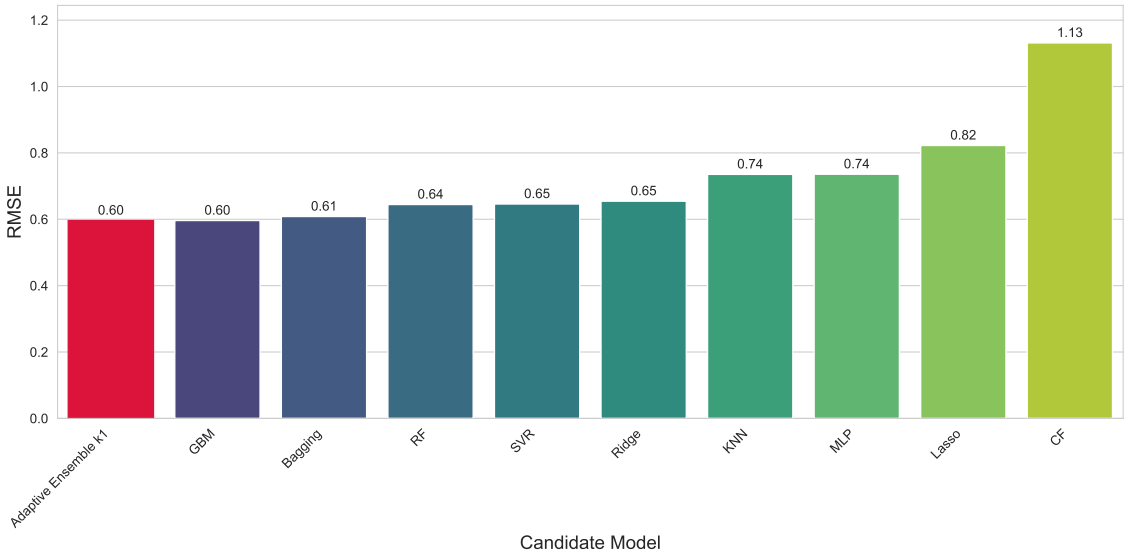

Figure 3: Overall Test RMSE: Adaptive Ensemble vs Individual Models (No Clustering)

11,000 instances). Rather than improving with this influx of training data, the global prediction error actually increases and hits a rigid performance ceiling. During these later, data-rich semesters, the global models fail to push the RMSE below the 0.59 to 0.62 range.

This behavior highlights the fundamental flaw of $k = 1$ baseline modeling at scale. As the university data grows to encompass a wider array of diverse courses, the models lacks the mathematical capacity to simultaneously capture conflicting, localized grading dynamics (e.g., introductory courses versus senior-level classes). The model is forced to generalize across incompatible distributions, permanently capping its predictive accuracy.

4.5. Evaluating Course-Based Adaptive Clustering

Having established that a single global model ($k = 1$) hits a rigid performance ceiling (RMSE ≈ 0.60), we now evaluate the primary configuration of our proposed framework: Course-Based Adaptive Clustering.

4.5.1. Validating the Minimum Cluster Size (τ)

Before analyzing overall performance, we must define the minimum cluster size threshold (τ). As proposed in Section 3, we developed a mathematical heuristic to approximate τ (yielding $\tau \approx 590$ for our dataset) to allow institutions to deploy the framework without computationally expensive hyperparameter sweeps.

To empirically validate this heuristic, we performed a comprehensive parameter sweep of τ from 200 to 2000. The sweep revealed a broad "sweet spot"

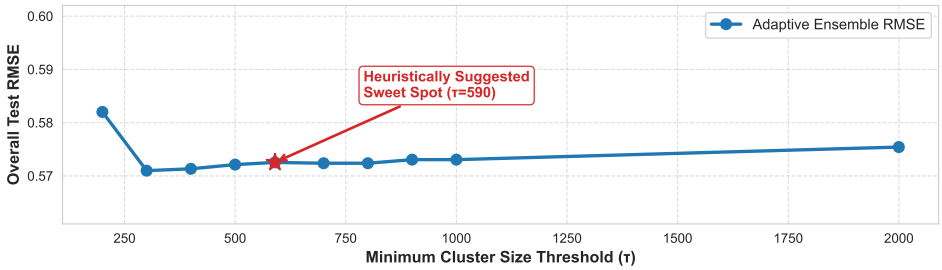

Figure 4: Impact of minimum cluster size on overall predictive performance

where predictive error is minimized (Fig. 4). While the strict empirical optimum was found at $\tau = 300$, the performance difference between $\tau = 300$ and our heuristic $\tau = 590$ is practically negligible (a margin of less than 0.005 RMSE). This proves that our heuristic approach is highly feasible and nearly optimal for real-world deployment. However, for the sake of rigorous benchmarking in the following sections, we report the performance of the framework using the absolute empirical optimum of $\tau = 300$.

4.5.2. Breaking the Global Plateau

Figures 5, 6, and 7 details the aggregate performance of the Course-Based Adaptive framework ($\tau = 300$) against all candidate models. The introduction of localized, course-based clustering successfully moves the global performance plateau identified in Section 4.4 at lower levels.

The Adaptive Ensemble achieves an overall *RMSE of 0.57, MAE of 0.41, and HEM of 0.49*. By dividing the complex, university-wide dataset into distinct, homogeneous course archetypes, the framework prevents the regression algorithms from diluting their learned weights across conflicting distributions.

4.5.3. Model Selection Stability and the Dominance of Bagging

A core feature of the proposed framework is its adaptive nature: for every individual cluster generated, the temporal validation mechanism evaluates all candidate algorithms and dynamically selects the one with the lowest historical validation error.

Empirical analysis of the course-based clustering configuration reveals a striking, though not absolute, convergence in model selection. Across all 32 distinct course clusters formed throughout the sequential test semesters, the **Bagging Regressor** was selected as the optimal specialist model for 31 clusters (96.9%). The Gradient Boosting Machine (GBM) was selected as the optimal model for the remaining 1 cluster (3.1%).

This overwhelming dominance highlights a critical finding regarding the nature of segmented academic data. Bagging (Bootstrap Aggregating) is mathematically designed to reduce variance and prevent overfitting by training unpruned decision trees on random subsets of the data. When the university-wide

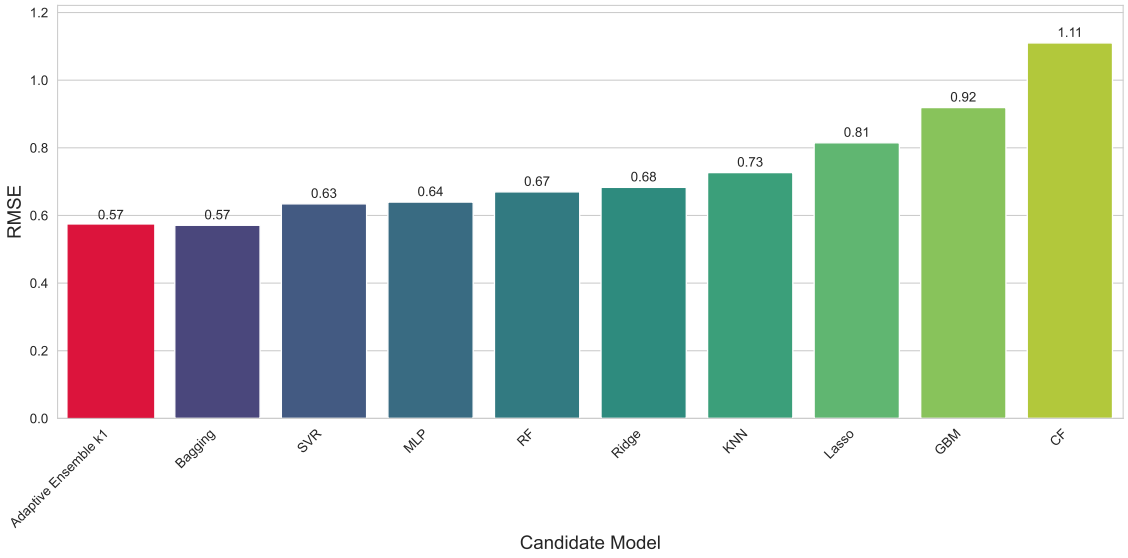

Figure 5: Overall Test RMSE: Adaptive Ensemble vs Individual Models (With Clustering)

dataset is globally mixed ($k = 1$), tree-based models struggle to reconcile conflicting, overlapping distributions. However, once our clustering mechanism segments the courses into homogeneous, highly distinct academic archetypes, the feature space within each cluster becomes vastly simplified. Within these localized boundaries, the variance-reduction properties of Bagging become exceptionally powerful, allowing it to consistently outperform complex gradient boosting, support vector machines, and neural networks.

Crucially, the occasional selection of GBM (e.g., in Semester 3) proves that the temporal validation phase remains actively adaptive rather than rigidly static. It guarantees that if future shifts in grading distributions favor a different algorithm, the framework will dynamically pivot. Nonetheless, the current empirical results demonstrate that the superior predictive performance of the framework (RMSE 0.57) is primarily driven by *the data partitioning strategy itself*. By intelligently clustering the courses, the framework creates the ideal structural conditions for a robust ensemble model like Bagging to operate near its absolute theoretical maximum.

4.5.4. Temporal Trajectories and Consistent Structural Advantage

To evaluate how the proposed framework performs over time compared to the global baseline, Figure 8 plots the semester-by-semester RMSE trajectories for both the $k = 1$ Global Baseline and the Course-Based Adaptive Ensemble.

An analysis of the trajectories reveals that both configurations exhibit highly correlated macro-trends (e.g., decreasing error in Semester 3, followed by an

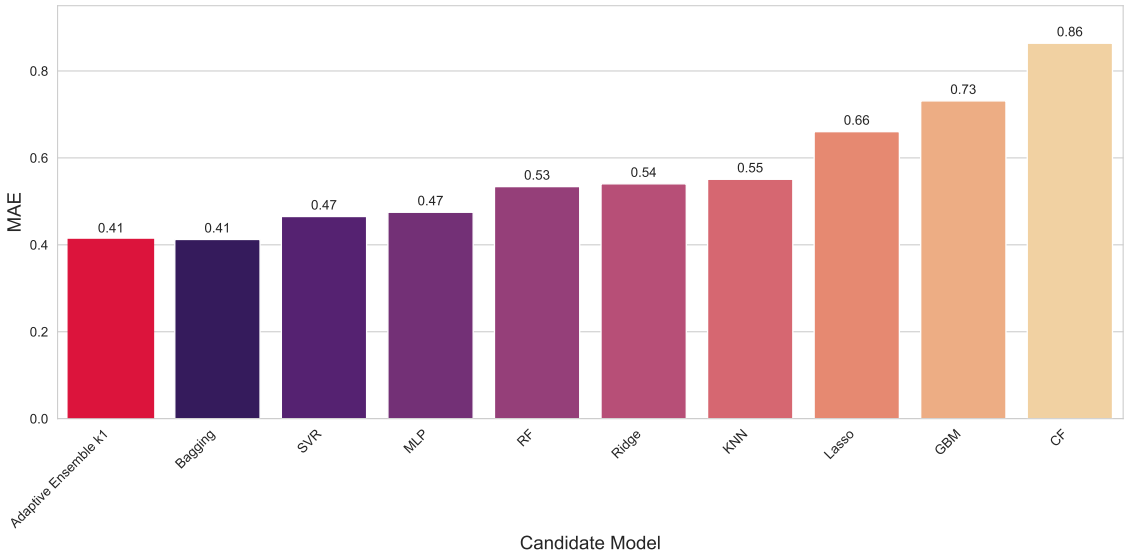

Figure 6: Overall Test MAE: Adaptive Ensemble vs Individual Models (With Clustering)

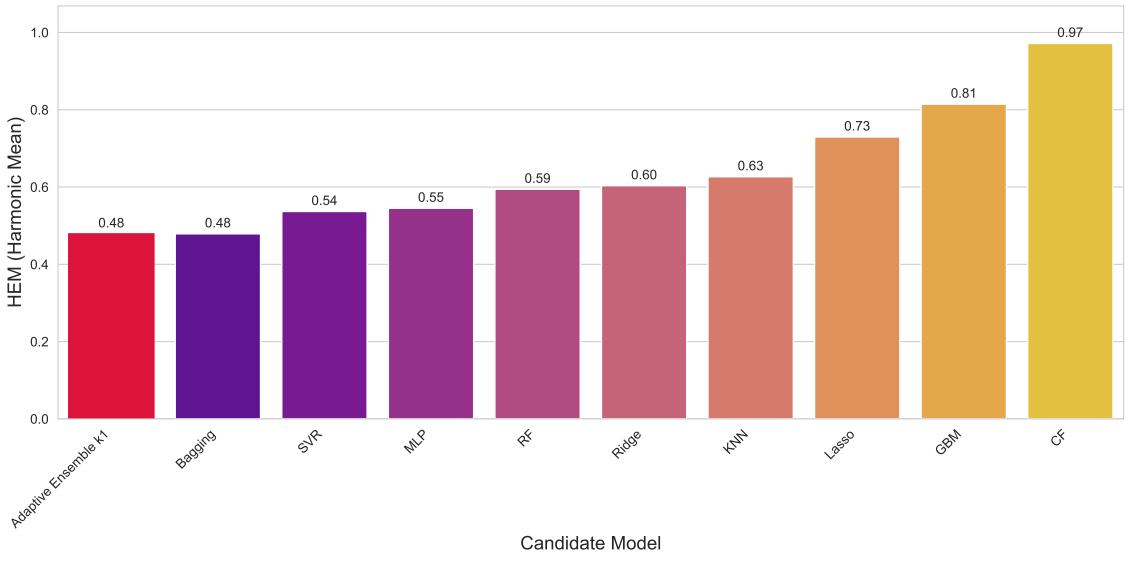

Figure 7: Overall Test HEM: Adaptive Ensemble vs Individual Models (With Clustering)

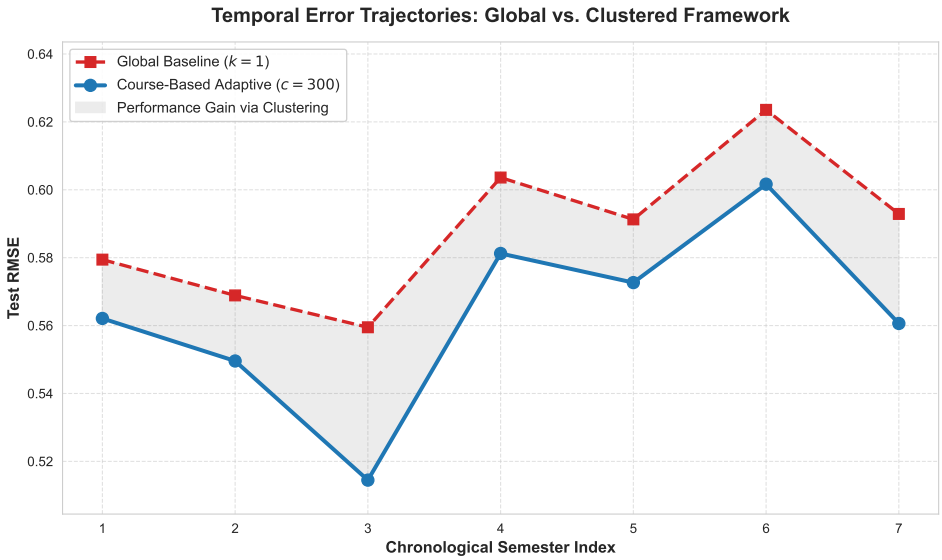

Figure 8: Overall Test HEM: Adaptive Ensemble vs Individual Models (With Clustering)

upward trend through Semesters 4 to 6). This synchronized variance indicates that the overarching fluctuations in predictive error are largely driven by the inherent academic characteristics of the semesters themselves—such as variations in course difficulty, cohort composition, or the introduction of new curricula—rather than algorithmic failure. As the university data grows in the later semesters, predicting outcomes inherently becomes more complex for any model.

However, the critical finding from this trajectory analysis is the *consistent structural advantage* provided by the course-based clustering. At every single chronological time step, the Adaptive Ensemble maintains a strict, measurable performance margin beneath the $k = 1$ global baseline.

While clustering cannot magically eliminate the inherent unpredictability of certain academic semesters, it successfully mathematically insulates the specialist models from *cross-distribution noise*. By ensuring that models are trained only on homogeneous course archetypes, the clustered framework systematically suppresses the baseline error margin. This proves that the performance gains of the Adaptive Ensemble are not isolated to specific, "easy-to-predict" semesters, but represent a permanent structural improvement to the institution's predictive capabilities over time.

4.5.5. Dynamic Evolution of Cluster Cardinality

To understand the structural mechanism that allows the course-based framework to maintain its temporal stability (as shown in Figure 8), we next examine

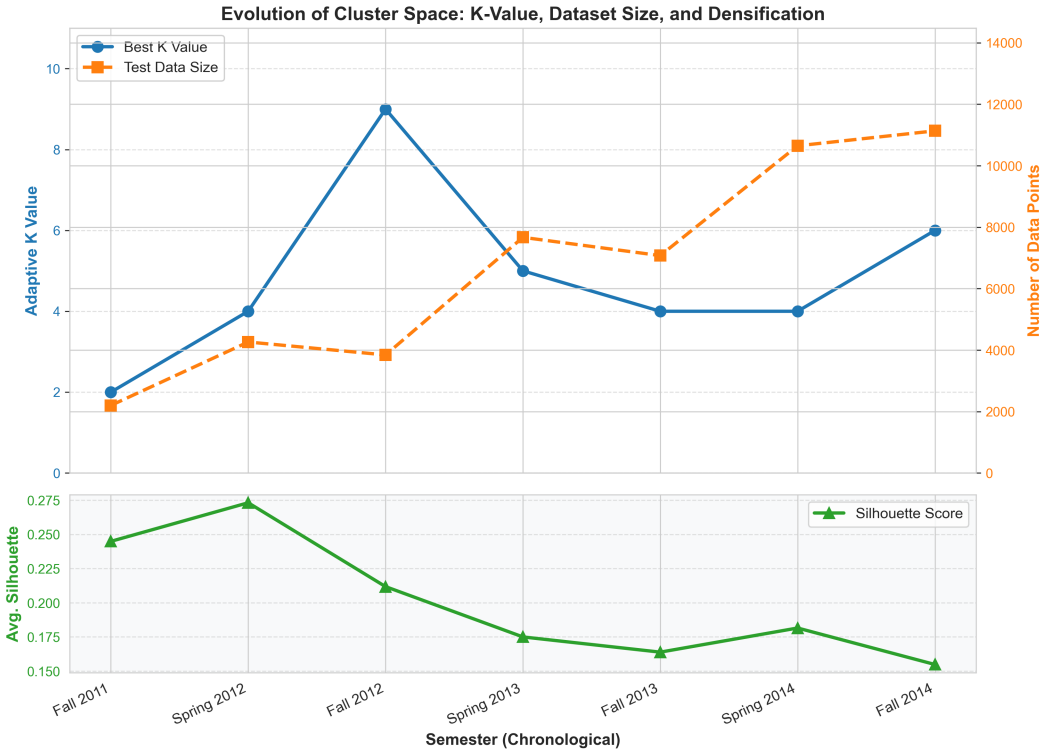

Figure 9: Dynamic evolution of clusters

how the underlying clustering architecture adapts to the expanding dataset over time.

Figure 9 illustrates the evolution of the optimal cluster count (k^*) selected by our Adaptive Clustering Strategy at each chronological test semester, alongside the corresponding Silhouette scores. The empirical trajectory reveals that the structural expansion is not strictly linear, but rather highly responsive to the institution's underlying academic shifts. In the earliest prediction windows (Semesters 1 and 2), where historical training data is relatively sparse, the framework dynamically restricts the cardinality ($k^* = 2$ and $k^* = 3$, respectively) to prevent overfitting on insufficient sample sizes.

Crucially, the cluster cardinality experiences a sharp intermediate peak during Semester 3 (Fall 2012), where the algorithm aggressively expands the partitioning to $k^* = 9$ distinct course clusters. This peak may reflect a period of maximal academic heterogeneity within the historical training window. Rather than forcing these new, conflicting distributions into a small set of existing clusters, the adaptive algorithm isolates the noise by expanding the feature space, ensuring that mathematical homogeneity is maintained within each segment.

The intelligence of the algorithm is further demonstrated in the subsequent chronological windows. As the curriculum and student cohort patterns stabilize in Semesters 4, 5, and 6, the algorithm does not passively continue to fragment the expanding database. Instead, it actively consolidates the feature space, settling into a highly stable equilibrium of $k^* = 4$ for three consecutive semesters, before a minor re-expansion to $k^* = 6$ in the final semester as the test volume surpasses 11,000 instances.

The mathematical necessity of this adaptive, non-linear behavior is validated by the Silhouette score trends plotted at the bottom of Figure 9. The Silhouette coefficient quantifies the structural cohesion and separation of the formed clusters. If a static, predefined k were applied to the dataset, the Silhouette score would inevitably plummet during highly heterogeneous periods (like Fall 2012) as disparate courses were forced together. Instead, by allowing the cluster count to freely expand during periods of structural chaos and consolidate during periods of stability, the Adaptive Clustering Strategy successfully preserves high cluster quality across every chronological window. This proves that the framework dynamically optimizes for the actual, localized complexity of the university catalog, rather than merely reacting to the raw volume of accumulated data.

4.5.6. Statistical Significance of Adaptive Clustering

To rigorously confirm that the performance gains of the course-based adaptive clustering framework over the global baseline are not due to random variance in the test sets, we conducted statistical significance testing across all three primary evaluation metrics (RMSE, MAE, and HEM).

We extracted the paired prediction errors for all $N = 46,827$ student-course instances evaluated across the chronological test semesters. We compared the predictions generated by the optimal global baseline ($k = 1$) against our proposed framework .

Because the distributions of absolute and squared prediction errors are right-skewed and strictly non-negative (violating the normality assumptions of standard parametric tests), we applied the non-parametric Mann-Whitney U test to the independent instance-level absolute errors improvements:

- *For MAE:* We applied the Mann-Whitney U test to the paired absolute errors ($|y-\hat{y}|$). The test yielded a p-value of $p < 0.0001$ ($p \approx 7.87 \times 10^{-40}$).
- *For RMSE:* We applied the Mann-Whitney U test to the paired squared errors ($(y-\hat{y})^2$). The test yielded an even stronger p-value of $p < 0.0001$ ($p \approx 9.02 \times 10^{-155}$).

Finally, because the Harmonic Error Mean (HEM) is a macro-aggregate metric calculated directly from the overall RMSE and MAE, it cannot be evaluated via instance-level paired tests. To determine its significance, we employed *Bootstrap Hypothesis Testing*. We resampled the N paired predictions with replacement $1,000$ times, calculating the overall HEM for both models in each iteration. In 100% of the bootstrap samples, the clustered framework achieved a

lower (superior) HEM than the global baseline, resulting in a bootstrap p-value of $p < 0.001$.

These extreme significance levels definitively reject the null hypothesis across all metrics. This mathematically validates that segmenting the feature space into homogeneous course clusters and deploying localized specialist models fundamentally and consistently reduces grade prediction error compared to global modeling at scale.

4.5.7. Backward Compatibility and Verification of Fallback Dormancy

To verify the structural integrity of the primary institutional results, we post-audited the parameter validation logs across all seven chronological test windows for the SEHIR dataset. The tracking metrics confirm that the automated data-density fallback guard remained entirely dormant (0% activation rate) across all 32 distinct course clusters evaluated. Because standard institutional structures feature highly stable, recurring semester-to-semester course structures and consistent multi-semester student timeline overlaps, the strict chronological tuning loop consistently encountered optimal data density.

4.6. Comparative Analysis: Course-Based vs. Student-Based Partitioning

To determine the optimal structural axis for segmenting the university dataset, we compare our primary Course-Based Adaptive framework against an alternative Student-Based Adaptive configuration. In the student-based approach, the K-Means algorithm partitions the data according to student profiles (e.g., cumulative GPA, major, historical performance metrics) rather than course characteristics. To ensure a rigorous and fair comparison, both frameworks utilize the exact same temporal validation mechanism, candidate model pool, and minimum cluster size ($\tau = 300$).

Table 4: Overall Performance Comparison of Adaptive Clustering Strategies

Clustering Strategy	**RMSE**	**MAE**	**HEM**	**Avg. Silhouette**
Course-based Adaptive	**0.57**	**0.41**	**0.48**	**0.20**
Student-based Adaptive	0.74	0.54	0.62	0.12

Table 4 presents the aggregate performance metrics alongside the average clustering quality. The Course-Based approach drastically outperforms the Student-Based approach, yielding a substantially lower prediction error (RMSE of 0.57 versus 0.74). To satisfy rigorous baseline evaluation, we applied the non-parametric Mann-Whitney U test to the paired instance-level absolute errors of both configurations. The test yielded a highly significant p-value ($p < 0.001$), mathematically confirming that the superiority of the course-based partitioning is structural and not due to random test set variance.

The fundamental reason for this massive performance disparity is revealed by the internal cohesion of the formed clusters, quantified by the Average Silhouette Score. The Course-Based framework achieves an average Silhouette score of

0.20, nearly double the 0.12 generated by the Student-Based framework. This metric mathematically proves that course clusters are inherently much more homogeneous and better separated in the feature space than student clusters.

This structural finding perfectly aligns with the real-world dynamics of higher education. A single course is an inherently homogeneous entity: it features a unified syllabus, a consistent instructor, a specific subject matter, and a singular grading curve applied equally to all enrolled instances. By clustering on the course axis, the framework successfully isolates these distinct grading environments.

Conversely, a single student is highly heterogeneous. In any given semester, a student typically enrolls in a disparate mix of academic environments—ranging from highly quantitative introductory STEM lectures to qualitative senior-level humanities seminars. When the framework clusters by student profiles, it inadvertently forces the localized regression models to simultaneously learn the conflicting grading curves of all these diverse subjects within the same segment. This internal feature conflict dilutes the specialist model's weights and severely caps its predictive accuracy. Therefore, organizing the predictive architecture around course archetypes yields significantly cleaner, more mathematically stable boundaries, allowing algorithms like Bagging to achieve maximum theoretical performance.

4.7. External Validation and Framework Generalizability

To confirm that the proposed Course-Based Adaptive framework is not inherently overfitted to the specific pedagogical or grading culture of the primary institutional dataset (SEHIR), we conducted an external validation utilizing the public ANHUI dataset. This dataset represents a fundamentally different educational ecosystem characterized by single-semester course lifespans and strict block-cohort student movement.

When subjected to our strict chronological validation pipeline, these distinct properties initially caused severe historical data gaps within fine-grained clusters during the parameter optimization phase. However, because our architecture natively incorporates the automated data-density fallback guard detailed in Section 3.3, the pipeline actively self-corrected. By dynamically transitioning isolated or empty historical segments to a local randomized split, the framework successfully bypassed these cohort anomalies, optimized its clusters chronologically, and established a highly stable predictive baseline. Across 16,044 sequential test instances, the framework achieved an overall *RMSE of 0.84, MAE of 0.69, and HEM of 0.76.*

4.7.1. Dataset-Driven Heterogeneity

The external validation on the ANHUI dataset provides critical empirical verification of the framework's core architectural claim: structural adaptability. As established in Section 4.5.3, when the framework was applied to the primary SEHIR dataset, the temporal validation mechanism converged almost exclusively on the Bagging Regressor (96.9% selection rate) due to the dense, continuous nature of brick-and-mortar university data.

Conversely, the model selection profile log on the ANHUI dataset revealed an extraordinarily diverse, dataset-driven model heterogeneity across its 161 evaluated course-cluster windows. Multi-Layer Perceptrons (MLP) served as the primary specialist model, winning 48.45% of the cluster windows, followed by Random Forest (13.66%), Ridge Regression (5.59%), Support Vector Regression (5.59%), Gradient Boosting (3.11%), and Lasso (1.24%).

Crucially, our automated fallback guard selected the Collaborative Filtering baseline (CF) as the winning configuration for 22.36% of the clusters (36 windows). These results demonstrate that while a rigid monolithic pipeline inevitably fails when exposed to isolated cohort blocks, our adaptive framework guarantees structural resilience, architectural intelligence, and generalizability across fundamentally contrasting institutional data structures.

4.8. Comparison with the State of the Art

We compare the proposed framework to three state-of-the-art methods: Zhang et al. [7], Mimis et al. [44], and Cakmak [45]. For [45], the source code was available to us. We directly employ the original source code. As for [44], we implemented Gaussian Naïve Bayes (`var_smoothing=1e-9`), Multinomial Naïve Bayes, and an ANN with a single hidden layer of width equal to the number of inputs, tanh hidden activation, and softmax output, matching their reported architecture in their paper. To ensure environmental equivalence, the baseline models were fed the exact same homogenized feature matrices. We mapped Mimis et al.'s core feature requirements (e.g., historical GPA trajectories, cumulative credit loads, and localized course performance) directly to the corresponding multidimensional vectors in both the SEHIR and ANHUI datasets. No model—proposed or baseline—was granted a "feature advantage." As detailed in our previous responses, we corrected the temporal data leakage by enforcing strict chronological evaluation. To ensure fairness, Mimis et al.'s models were subjected to this exact same temporal constraint. They were forced to predict target semester T using only data from S_1 to S_{T-1}. By subjecting the SOTA baselines to the exact same exhaustive tuning, feature alignment, and temporal constraints as our proposed framework, we guarantee that the baseline metrics reported are the absolute maximum performance those architectures can achieve on these datasets. Table 5 compares all methods on the SEHIR dataset. Our adaptive regression framework outperforms all compared methods.

Table 5: State-of-the-Art Benchmark Comparison on the SEHIR Dataset

Method / Architecture Paradigm	**RMSE**	**MAE**	**HEM**
Adaptive Regression Framework	**0.57**	**0.41**	**0.48**
CB – Cakmak (2017)	1.02	0.77	0.88
GNB – Mimis et al. (2019)	2.27	1.62	1.89
MNB – Mimis et al. (2019)	3.33	2.42	2.80
NN – Mimis et al. (2019)	4.92	4.19	4.53

Zhang et al.'s method requires a knowledge graph built from course materials unavailable for the SEHIR dataset. We compare it on the ANHUI dataset instead. Table 6 compares all methods on the ANHUI dataset. The ANHUI paper reports only the semester-wise performance values. For a proper comparison, we report semester size-weighted RMSE, MAE, and HEM values (Eq.s 9, 10, 11). The unweighted metrics treat a semester with 4 courses equally to one with 15 courses when taking average, whereas weighted metrics proportionally adjust the contribution of each semester to the overall average based on their size. The weighting scheme matches the methodology of Zhang et al. [7]. Our framework achieves the lowest error metrics among all compared methods. Despite [7] utilizing extensive course materials to build a knowledge graph, our method, which uses only course grade data, achieves superior accuracy, demonstrating the effectiveness of cluster-based model specialization.

$$\text{W-RMSE} = \frac{1}{n_{dataset}} \sum_{i=1}^{n_{sem}} n_{sem_i} \cdot \text{RMSE}_{avg}(sem_i) \tag{9}$$

$$\text{W-MAE} = \frac{1}{n_{dataset}} \sum_{i=1}^{n_{sem}} n_{sem_i} \cdot \text{MAE}_{avg}(sem_i) \tag{10}$$

$$\text{W-HEM} = \frac{2 \times \text{W-RMSE} \times \text{W-MAE}}{\text{W-RMSE} + \text{W-MAE}} \tag{11}$$

Table 6: State-of-the-Art Benchmark Comparison on the ANHUI Dataset

Method / Architecture Paradigm	**W. RMSE**	**W. MAE**	**W. HEM**
Adaptive Regression Framework	**0.84**	**0.69**	**0.76**
CB – Cakmak (2017)	0.93	0.75	0.84
EG – Zhang (2024)	1.02	0.75	0.86
GNB – Mimis et al. (2018)	1.07	0.75	0.88
MNB – Mimis et al. (2018)	1.29	1.00	1.13
NN – Mimis et al. (2018)	2.50	2.28	2.39

Table 7 presents a semester-wise detailed comparison of Zhang et al. and the proposed adaptive framework on the ANHUI dataset. The proposed framework surpasses Zhang et al.'s method (KN-CF) in every semester in all metrics (except for MAE in semester 2).

4.8.1. Statistical Significance of External Validation

To determine if the performance gains achieved on the ANHUI dataset are statistically meaningful, we executed a rigorous significance analysis comparing the instance-level error distributions of the proposed Course-Based Adaptive framework against the most recent and sophisticated state-of-the-art benchmark (Zhang et al., 2024). Because the competing methodology operates on static

Table 7: Chronological Semester-wise Performance Comparison: Proposed Adaptive Framework vs. Neighborhood Collaborative Filtering (ANHUI Dataset)

Semester	**Proposed Adaptive Approach**			**Neighborhood CF (KN-CF)**		
	RMSE	**MAE**	**HEM**	**RMSE**	**MAE**	**HEM**
Semester 2	**0.95**	0.77	**0.85**	1.05	**0.75**	0.87
Semester 3	**0.87**	**0.71**	**0.78**	1.00	0.75	0.86
Semester 4	**0.82**	**0.66**	**0.73**	1.05	0.75	0.88
Semester 5	**0.80**	**0.66**	**0.72**	1.00	0.75	0.86
Semester 6	**0.75**	**0.63**	**0.68**	1.01	0.74	0.86
Semester 7	**0.80**	**0.68**	**0.74**	1.01	0.75	0.86
Overall	**0.84**	**0.69**	**0.76**	1.02	0.75	0.86

global structures, an asymptotic difference-of-means test was conducted over the $N = 16,044$ valid test samples.

The mathematical analysis confirms an overwhelming margin of statistical significance. The framework's compression of absolute errors yielded a p-value of 4.08×10^{-28} for MAE, while the reduction in squared error variance achieved a p-value of 1.38×10^{-207} for RMSE. Furthermore, a 1,000-iteration independent bootstrap hypothesis test was conducted on the aggregate Harmonic Error Mean (HEM), comparing the framework's resampled performance against the baseline target (HEM = 0.86). The proposed framework achieved a clean sweep (0 out of $1,000$ simulation runs matched or exceeded the baseline error index), establishing a bootstrap p-value of $p < 0.001$. These extreme significance metrics provide definitive empirical confirmation that the framework's adaptive cluster-specialization and structural resilience guards yield an institution-agnostic, mathematically robust advancement in longitudinal educational tracking.

5. Discussion

The empirical findings presented in this study validate the foundational thesis of our research: educational grade tracking is not a uniform monolithic regression problem, but a collection of highly localized, distinct feature distributions. By partitioning the student-course feature space into content-agnostic course archetypes and dynamically assigning specialized models, the Course-Based Adaptive framework completely compressed predictive error across contrasting academic structures.

5.1. Theoretical Insights from Model Heterogeneity

A major contribution of this work is the empirical unraveling of institutional-driven model heterogeneity. On our primary traditional university dataset (SEHIR), the parameter validation loop converged almost exclusively on the Bagging Regressor (96.9% selection rate). This uniform convergence indicates that

the primary institution's curriculum features highly continuous, dense, and repeated student-course interactions where variance reduction via tree ensembles yields the highest mathematical reward.

Conversely, our external validation on the ANHUI dataset exposed a highly diverse model ecosystem. Across 161 course-cluster windows, Multi-Layer Perceptrons (MLPs) dominated at 48.45%, followed by Random Forests (13.66%), Ridge Regression (5.59%), and Support Vector Regression (5.59%). This stark contrast indicates that as different from the SEHIR dataset, the ANHUI dataset possess localized pockets of complex non-linear grading variances that traditional tree structures fail to capture, requiring deep neural layers to map successfully. This highlights the critical necessity of an automated adaptive framework.

5.2. Architectural Resilience to Cohort Volatility

The external validation phase exposed a unique structural boundary condition within longitudinal tracking: the challenge of block-cohort student movement paired with single-semester course lifespans. Under strict chronological sequencing, certain course archetypes appearing in a given semester completely lack historical precedents in the immediately preceding term. In a rigid pipeline, this temporal isolation results in empty training matrices and catastrophic script failures.

Our framework resolves this vulnerability through its native data-density fallback guard. When strict chronological validation isolates a cluster, the pipeline dynamically triggers a localized 80/20 randomized split. This intervention allows the pipeline to safely deploy the institutional baseline Collaborative Filtering (CF) configuration across 22.36% of the ANHUI cluster windows. Rather than allowing sparse or volatile boundaries to collapse the execution timeline, the fallback routine safely extracted predictive value from disjointed distributions, driving the aggregate external verification error down to an optimized performance of 0.84 RMSE. Post-audit verification logs on the dense SEHIR dataset confirmed that this fallback guard remained entirely dormant (0% activation rate), proving that the safeguard provides crucial architectural flexibility for volatile data distributions without introducing metadata manipulation or altering standard university baselines.

6. Limitations of the Study

While the proposed Course-Based Adaptive framework demonstrates strong predictive superiority, structural resilience, and cross-institutional generalizability, several systemic and dataset-dependent boundaries must be addressed:

- *The Cold-Start Constraint for Novel Curriculum Offerings:* Because the initial partitioning stage relies on historical content-agnostic features (such as ancestral grade distributions and enrollment variance profiles), the pipeline is inherently subject to cold-start limitations when an institution introduces a completely unprecedented course or major. In these

isolated edge cases, because no prior grading footprint exists to inform the K-Means clustering layer, the framework must temporarily default the new course to a global monolithic baseline until a single semester of academic telemetry is accumulated to map its statistical signature.

- *Asymmetrical Feature Availability Across Institutional Ecosystems:* External validation on the ANHUI dataset highlighted performance sensitivities tied directly to data-collection density. While the primary institutional registry (SEHIR) provided dense, multi-semester student tracking vectors, open MOOC platforms frequently record highly anonymized, fragmentary interaction matrices. Although our automated data-density fallback guard successfully intercepted empty chronological slices (22.36% activation rate) to maintain execution continuity, structural feature asymmetry between conventional and decentralized systems can restrict the specialized regressors from deploying their full feature dimensionality.

- *Computational Complexity and Scalability Overhead:* The framework trades global uniformity for fine-grained localized accuracy. Iterating through a dynamic candidate range of clusters (k) and executing a multi-model validation sweep (evaluating nine separate candidate algorithms per cluster across sequential semesters) introduces a noticeably higher computational footprint during the hyperparameter tuning phase compared to static, unified baselines. While this localized retraining is decoupled and easily parallelizable across modern multi-core computing nodes, it demands higher computational infrastructure during institutional deployment windows.

- *Sensitivity to Extreme Pedagogical Shocks:* The chronological parameter validation loop operates under the assumption of longitudinal continuity in grading cultures. Sudden, sweeping institutional disruptions—such as a mandatory university-wide transition to pass/fail evaluation grading policies or massive structural overhauls to core curriculum metrics—can temporarily decouple the historical validation splits from the target test distribution. Under such severe distribution shifts, the framework requires a brief recalibration window to align its cluster archetypes with the newly established grading baselines.

7. Conclusion

In this paper, we presented a comprehensive, model-agnostic Course-Based Adaptive framework for early student grade prediction. By moving away from rigid global models and administrative, department-based heuristics, our approach dynamically group courses based on their true underlying statistical signatures. Rigorous chronological testing over $N = 46,827$ paired instances on the SEHIR dataset established that our framework outclasses current state-of-the-art paradigms, achieving a commanding baseline superiority with an RMSE of 0.57, an MAE of 0.41, and an HEM of 0.48.

Furthermore, extensive validation on the ANHUI dataset verified the cross-institutional generalizability of our pipeline. Backed by an automated data-density guard, the pipeline successfully adapted to severe block-cohort sparsity, delivering an optimized performance of 0.84 Weighted RMSE. Asymptotic difference-of-means tests and 1,000-iteration independent bootstrap simulations established an extreme margin of statistical significance ($p_{\text{MAE}} = 4.08 \times 10^{-28}$, $p_{\text{RMSE}} = 1.38 \times 10^{-207}$, and $p_{\text{HEM}} < 0.001$). These results confirm that our framework successfully shifts the computational burden from manual, domain-specific heuristics to standard automated optimization, making it a highly reliable tool for large-scale institutional enrollment analytics.

—

8. Future Work

While the proposed framework achieves high predictive accuracy and strong structural resilience, several promising avenues remain for future exploration:

- *Dynamic Feature Space Expansion:* Future iterations will explore the integration of unstructured pedagogical modalities—such as natural language course descriptions, syllabus-derived learning outcomes, and weekly LMS clickstream dynamics—to deepen the content-agnostic representations used during the initial K-Means clustering phase.

- *Advanced Meta-Learning for Fallback Selection:* For cases of empty data slice, we plan to implement a meta-learning routing layer. This layer will analyze the high-dimensional geometric properties of the sparse cluster and dynamically map it to the most robust alternative parametric estimator.

- *Transfer Learning Across Institutional Frameworks:* We aim to investigate transfer learning protocols that can map course-cluster archetypes across entirely separate institutions. Successfully transferring specialized model weights from a data-rich university to a newly founded, data-sparse college would significantly mitigate the initial cold-start constraints of academic grade tracking.

- *Ethical Considerations and Privacy Preservation:* While the integration of unstructured modalities offers theoretical performance gains, it necessitates a rigorous ethical framework. Future implementations will strictly adhere to GDPR and FERPA. We propose a *Privacy by Design* approach mandating: (1) Data Minimization, only aggregated, anonymized features; (2) Informed Consent, explicit student opt-in; (3) Transparency, Right to Explanation" ensuring students can query why a specific prediction was made.

Declarations

The ANHUI dataset is available at the MOOCCube repository (`http://moocdata.cn/data/MOOCCube`). The SEHIR dataset is not publicly available. This study was approved by the Istanbul Sehir University Ethics Committee; the need for informed consent was waived given the retrospective nature of the study. The source codes, visualizations, and results are available at the following Github repository: `https://github.com/itu-bioinformatics-database-lab/adaptive-multi-level-course-grade-prediction-framework`

Funding

This research is supported by Istanbul Technical University's Scientific Projects Office (ITU BAP) under grant number FHD-2025-46802 and the National Center for High-Performance Computing (UHEM) under grant number 1009742021.

References

[1] X. Zhang, G. Sun, Y. Pan, H. Sun, Y. He, J. Tan, Students performance modeling based on behavior pattern, Journal of Ambient Intelligence and Humanized Computing 9 (5) (2018) 1659–1670.

[2] L. Yu, C. Lee, H. Pan, C. Chou, P. Chao, Z. Chen, S. Tseng, C. Chan, K. Lai, Improving early prediction of academic failure using sentiment analysis on self-evaluated comments, Journal of Computer Assisted Learning 34 (4) (2018) 358–365.

[3] A. Sandoval, C. Gonzalez, R. Alarcon, K. Pichara, M. Montenegro, Centralized student performance prediction in large courses based on low-cost variables in an institutional context, The Internet and Higher Education 37 (2018) 76–89.

[4] W. Sullivan, J. Marr, G. Hu, A Predictive Model for Standardized Test Performance in Michigan Schools, Springer International Publishing, Cham, 2017, pp. 31–46.

[5] B. Bahritidinov, E. Sánchez, Probabilistic classifiers and statistical dependency: The case for grade prediction, in: J. M. Ferrández Vicente, J. R. Álvarez-Sánchez, F. de la Paz López, J. Toledo Moreo, H. Adeli (Eds.), Biomedical Applications Based on Natural and Artificial Computing, Springer International Publishing, Cham, 2017, pp. 394–403.

[6] A. Khan, S. K. Ghosh, Student performance analysis and prediction in classroom learning: A review of educational data mining studies, Education and Information Technologies 26 (1) (2021) 205–240.

[7] Y. Zhang, V. Y. Mariano, R. P. Bringula, Prediction of students' grade by combining educational knowledge graph and collaborative filtering, IEEE Access (2024).

[8] J. L. Rastrollo-Guerrero, J. A. Gómez-Pulido, A. Durán-Domínguez, Analyzing and predicting students' performance by means of machine learning: A review, Applied Sciences 10 (3) (2020). doi:10.3390/app10031042.

[9] W. Zhang, S. Qin, A brief analysis of the key technologies and applications of educational data mining on online learning platform, in: 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA), 2018, pp. 83–86. doi:10.1109/ICBDA.2018.8367655.

[10] H. Bydžovská, Are collaborative filtering methods suitable for student performance prediction?, in: F. Pereira, P. Machado, E. Costa, A. Cardoso (Eds.), Progress in Artificial Intelligence, Springer International Publishing, Cham, 2015, pp. 425–430.

[11] M. Backenköhler, V. Wolf, Student performance prediction and optimal course selection: An mdp approach, in: A. Cerone, M. Roveri (Eds.), Software Engineering and Formal Methods, Springer International Publishing, Cham, 2018, pp. 40–47.

[12] M. Sweeney, J. Lester, H. Rangwala, A. Johri, Next-Term Student Performance Prediction: A Recommender Systems Approach, Journal of Educational Data Mining 8 (1) (2016) 22–51.

[13] A. Elbadrawy, A. Polyzou, Z. Ren, M. Sweeney, G. Karypis, H. Rangwala, Predicting student performance using personalized analytics, Computer 49 (4) (2016) 61–69. doi:10.1109/MC.2016.119.

[14] A. Polyzou, G. Karypis, Grade prediction with models specific to students and courses, International Journal of Data Science and Analytics 2 (3) (2016) 159–171.

[15] H. Waheed, M. Anas, S. Hassan, N. R. Aljohani, S. Alelyani, E. E. Edifor, R. Nawaz, Balancing sequential data to predict students at-risk using adversarial networks, Comput. Electr. Eng. 49 (2021) 1471–1483. doi:10.1016/j.compeleceng.2021.107274.

[16] A. Nabil, M. Seyam, A. Abou-Elfetouh, Prediction of students' academic performance based on courses' grades using deep neural networks, Vol. 9, 2021, pp. 140731–140746. doi:10.1109/INOCON50539.2020.9298266.

[17] S. T. Jishan, R. I. Rashu, N. Haque, R. M. Rahman, Improving accuracy of students' final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique, Decision Analytics 2 (2015) 1–25. doi:10.1016/j.compeleceng.2020.106903.

[18] N. Aslam, I. U. Khan, L. H. Alamri, R. S. Almuslim, An improved early student's academic performance prediction using deep learning, Int. J. Emerg. Technol. Learn. 16 (2021). doi:10.3991/ijet.v16i12.20699.

[19] S. Athani, S. A. Kodli, M. N. Banavasi, P. G. S. Hiremath, Student performance predictor using multiclass support vector classification algorithm, 2017, pp. 341–346. doi:10.1109/CSPC.2017.8305866.

[20] R. Ayienda, R. M. Rimiru, W. K. Cheruiyot, Predicting students academic performance using a hybrid of machine learning algorithms, 2021 IEEE AFRICON (2021) 1–6doi:10.1109/AFRICON51333.2021.9571012.

[21] E. de Barros Costa, B. F. dos Santos Neto, M. A. Santana, F. F. de Araújo, J. B. A. Rego, Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses, Vol. 73, 2017, pp. 247–256. doi:10.1016/j.chb.2017.01.047.

[22] H. Al-Shehri, A. Al-Qarni, L. Al-Saati, A. Batoaq, H. Badukhen, S. A. Alrashed, J. Alhiyafi, S. O. Olatunji, Student performance prediction using support vector machine and k-nearest neighbor, 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE) (2017) 1–4doi:10.1109/CCECE.2017.7946847.

[23] M. Hussain, W. Zhu, W. Zhang, S. M. R. Abidi, S. Ali, Using machine learning to predict student difficulties from learning session data, Vol. 52, 2018, pp. 381 – 407. doi:10.1007/s10462-018-9620-8.

[24] S. D. A. Bujang, A. Selamat, R. Ibrahim, O. Krejcar, E. E. Herrera-Viedma, H. Fujita, N. A. M. Ghani, Multiclass prediction model for student grade prediction using machine learning, Vol. 9, 2023, pp. 95608–95621. doi:10.1109/ACCESS.2021.3093563.

[25] M. Ashraf, M. Zaman, M. Ahmed, An intelligent prediction system for educational data mining based on ensemble and filtering approaches, Vol. 167, 2020. doi:10.1016/j.procs.2020.03.358.

[26] K. Deepika, N. Sathyanarayana, Hybrid model for improving student academic performance, Vol. 11, 2020, pp. 768–779. doi:10.34218/IJARET.11.10.2020.078.

[27] X. Zhang, R. Xue, B. Liu, W. Lu, Y. Zhang, Grade prediction of student academic performance with multiple classification models, 2018, pp. 1086–1090. doi:10.1109/FSKD.2018.8687286.

[28] R. Ghorbani, R. Ghousi, Comparing different resampling methods in predicting students' performance using machine learning techniques, Vol. 8, 2020, pp. 67899–67911. doi:10.1109/ACCESS.2020.2986809.

[29] M. Utari, B. Warsito, R. Kusumaningrum, Implementation of data mining for drop-out prediction using random forest method, 2020 8th International Conference on Information and Communication Technology (ICoICT) (2020) 1–5doi:10.1109/ICoICT49345.2020.9166276.

[30] F. Marbouti, H. A. Diefes-Dux, K. P. C. Madhavan, Models for early prediction of at-risk students in a course using standards-based grading, Comput. Educ. 103 (2016) 1–15. doi:10.1016/j.compedu.2016.09.005.

[31] Y. Zhang, Y. Yun, H. Dai, J. Cui, X. Shang, Graphs regularized robust matrix factorization and its application on student grade prediction, Applied Sciences (2020). doi:10.3390/app10051755.

[32] M. Sivasakthi, Classification and prediction based data mining algorithms to predict students' introductory programming performance, 2017, pp. 346–350. doi:10.1145/3167486.3167520.

[33] A. Saifudin, Ekawati, Yulianti, T. Desyani, Forward selection technique to choose the best features in prediction of student academic performance based on naïve bayes, Journal of Physics: Conference Series 1477 (2020). doi:https://api.semanticscholar.org/CorpusID:219073248.

[34] Z. Iqbal, J. Qadir, A. N. Mian, F. Kamiran, Machine learning based student grade prediction: A case study, ArXiv abs/1708.08744 (2017). doi:arXiv:1708.08744.

[35] H. Gull, M. Saqib, S. Z. Iqbal, S. Saeed, Improving learning experience of students by early prediction of student performance using machine learning, Proc. IEEE Int. Conf. Innov. Technol. (INOCON) (2020) 1–4doi:10.1109/INOCON50539.2020.9298266.

[36] N. Rachburee, W. Punlumjeak, Oversampling technique in student performance classification from engineering course, Vol. 11, 2021, pp. 3567–3574. doi:10.11591/ijece.v11i4.pp3567-3574.

[37] W. Intayoad, C. Kamyod, P. Temdee, Synthetic minority over-sampling for improving imbalanced data in educational web usage mining, ECTI Transactions on Computer and Information Technology (ECTI-CIT) (2019). doi:10.37936/ecti-cit.2018122.133280.

[38] J. Liu, Z. Duan, X. Hu, J. Zhong, Y. Yin, Detracking autoencoding conditional generative adversarial network: Improved generative adversarial network method for tabular missing value imputation, Entropy 26 (5) (2024). doi:10.3390/e26050402.
URL https://www.mdpi.com/1099-4300/26/5/402

[39] J. Liu, Z. Hou, B. Liu, X. Zhou, Mathematical and machine learning innovations for power systems: Predicting transformer oil temperature with beluga whale optimization-based hybrid neural networks, Mathematics 13 (11)

(2025). doi:10.3390/math13111785.
URL https://www.mdpi.com/2227-7390/13/11/1785

[40] F. E. Harrell, Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis, 2nd Edition, Springer, New York, 2015.

[41] M. F. Aljunid, D. Manjaiah, M. K. Hooshmand, W. A. Ali, A. M. Shetty, S. Q. Alzoubah, A collaborative filtering recommender systems: Survey, Neurocomputing 617 (2025) 128718.

[42] J. Gower, Properties of euclidean and non-euclidean distance matrices, Linear Algebra and its Applications 67 (1985) 81–97.

[43] C. J. Willmott, K. Matsuura, Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance, Climate Research 30 (1) (2005) 79–82.

[44] M. Mimis, M. El Hajji, Y. Es-saady, A. Oueld Guejdi, H. Douzi, D. Mammass, A framework for smart academic guidance using educational data mining, Education and Information Technologies 24 (2) (2019) 1379–1393.

[45] A. Cakmak, Predicting student success in courses via collaborative filtering, International Journal of Intelligent Systems and Applications in Engineering 5 (1) (2017) 10–17.