Combining Random Forest, Neural Networks, And Association Rules For Student Grade Prediction And Course Recommendations

Sami Alghmadi
Dept. of Computer Science, College of Computing and Information, Al-Baha University, Saudi Arabia

This study proposes an ensemble-based approach for predicting university student performance and recommending optimal course combinations. The approach integrates Random Forest (RF), Long Short-Term Memory networks (LSTM) with Attention mechanisms, and Association Rule Mining (ARM) to address both individual academic forecasting and institutional course planning. RF is employed for effective feature selection and classification, while LSTM-A captures temporal patterns in students' academic trajectories. ARM is used to extract interpretable associations between course groupings and performance trends. The dataset contains detailed transcript and study plan records from 107 undergraduate students across 17 semesters. The experimental evaluation shows that the ensemble model achieves an accuracy of 82% and a macro-F1 score of 80%, outperforming traditional machine learning techniques. Additionally, the framework successfully identifies at-risk students with 85% accuracy in early semesters, supporting its potential use for academic advising and early intervention strategies.

Keywords: Educational Data Mining, Ensemble Learning, Student Performance Prediction, Association Rule Mining, Course Recommendation Systems, Random Forest, Long Short-Term Memory Networks, Early Intervention

1 INTRODUCTION

The rapid digitization of educational systems has led to an explosion of student academic data, offering unprecedented opportunities for data-driven decision-making in higher education. Educational Data Mining (EDM) has emerged as a pivotal field, focusing on extracting actionable insights from educational data to predict student performance, identify at-risk individuals, and optimize learning pathways. Student performance prediction enables early intervention strategies, such as personalized tutoring, course adjustments, or additional support resources, which can significantly reduce dropout rates, improve retention, and enhance overall academic outcomes. For instance, early identification of struggling students can allow advisors to intervene before failures occur [1, 2], potentially saving educational resources and improving student satisfaction. In most higher education programs, where cumulative knowledge builds across semesters, such predictions are particularly valuable, as failure in foundational courses can cascade into broader academic difficulties [3, 4].

However, traditional academic counseling methods often rely on subjective assessments by advisors or limited historical context, such as overall GPA, which may skip complex patterns like temporal grade trends over semesters or dependencies between course combinations. These oversights can lead to undesirable decisions, such as recommending course pairs that may lead to poor performance [5, 6]. Recent studies [2, 5, 6, 7, 8] have shown that multi-dimensional factors, including workload, course type, and sequential performance, play a crucial role in student success, indicating the need for more advanced analytical methods that involve a mixture of machine learning approaches. The students' academic history such as the previous grades and cumulative GPA is one of the most important factors that influences the students' performance [5, 7, 10]. Furthermore, demographic factors including socio-economic background and parental education level have shown notable correlations with results [5, 8]. In addition to course-specific attributes such as difficulty level, credit load, and prerequisite dependencies [6, 11], and temporal patterns, including grade progress across semesters and course sequencing effects, which contribute to the early discovery of students at academic risk [7, 8].

Machine learning approaches have shown considerable effectiveness in student performance prediction, with varying accuracy rates across different methodologies. For example, the study in [12] achieved a high accuracy (96%) employing Support Vector Machines (SVM), outperforming Decision Trees (DT), Naive Bayes (NB), and K-Nearest Neighbors (KNN)classifiers. A systematic review was conducted in [13] of 39 studies published between 2015-2021, identifying DTs, Artificial Neural Networks (ANN), SVM, KNN, Linear Regression (LR), and NB as the six most commonly employed models. While the work in [14] achieved 70-75% classification accuracy using six machine learning algorithms to predict final exam grades from midterm performance, highlighting the importance of early prediction approaches.

More recently, ensemble learning approaches have consistently demonstrated superior performance by combining multiple models. Several studies [1, 6, 15, 16, 17, 18] have demonstrated that ensemble methods combining diverse algorithms such as Naive Bayes (NB), SVM, MLP, and Logistic Regression (LR) significantly outperform individual classifiers. Malik et al. [6] introduced dynamic feature ensemble evolution for enhanced feature selection in student performance prediction, achieving improved accuracy through optimized feature selection combined with ensemble classifiers. In [19] the authors achieved classification accuracy of 84%-93% in predicting at-risk students from the Open University Learning Analytics dataset by deploying a deep artificial neural network on handcrafted features outperforming baseline LR and SVM models. The study in [20] proposed a deep ensemble learning method combining multiple Deep Belief Networks optimized by particle swarm optimization with reinforcement learning-based weighting, achieving an RMSE of 1.66, MAPE of 9.75%, and R^2 of 0.7430, significantly outperforming conventional ensemble approaches.

Despite these advances, research gaps persist in the integration of predictive modeling with course recommendation systems. While recommendation system techniques have been extensively applied to discover course relationships and generate actionable scheduling recommendations, most existing approaches focus either on performance prediction or course recommendation separately, without exploring the benefits of combining both techniques.

This study addresses these gaps by proposing ensemble-based approach that integrates Random Forest (RF), Long Short-Term Memory networks with attention mechanisms (LSTM-A), and Association Rule Mining (ARM) [21] for student performance prediction and course combination recommendation. This approach leverages the complementary strengths of three techniques, RF for feature selection and classification, LSTM-A for capturing temporal patterns in academic data across multiple semesters, and ARM for extracting course associations. The work is evaluated using a real-world dataset containing detailed transcript and study plan records from 107 undergraduate students across 17 semesters. The proposed ensemble model achieves 82% accuracy and 80% macro-F1 score, outperforming traditional approaches. Furthermore, the approach identifies at-risk students with 85% accuracy in early semesters, demonstrating practical utility for academic advising and early intervention strategies. In addition to predicting student outcomes, integration of ARM enables the system to recommend optimal course combinations.

The remainder of this paper is organized as follows: Section 2 reviews related work in machine learning approaches, ensemble learning methods, and association rule mining. Section 3 describes the proposed ensemble framework architecture and methodology. Section 4 presents experimental setup, dataset characteristics, evaluation metrics, and results analysis. Finally, Section 5 concludes the paper and provides the future work.

2 RELATED WORK

The following review examines existing work in three primary domains: machine learning approaches for student performance prediction, ensemble learning methods, and association rule mining for course recommendation systems.

A. Machine Learning Approaches

Multiple studies have demonstrated the effectiveness of traditional machine learning algorithms with varying accuracy rates. Waheed et al. [19] proposed a deep artificial neural network on handcrafted features extracted from Virtual Learning Environment clickstream data, achieving classification accuracy of 84%-93% in predicting at-risk students from the Open University Learning Analytics dataset outperforming baseline LR and SVM models. Ahmed [12] examined SVM, Decision Tree (DT), Naive Bayes (NB), and KNN classifiers, finding that SVM achieved optimal performance with 96% accuracy after parameter tuning, followed by DT with 93.4% accuracy. However, the high accuracy was achieved on a specific dataset with tuned parameters. Alsariera et al. [13] conducted a systematic review of 39 studies published between 2015-2021, identifying DTs, ANNs, SVMs, KNN, LR, and NB as the six most commonly employed models, with academic, demographic, internal assessment, and family-personal attributes as the most predominant predictive features. Yağcı [14] achieved 70-75% classification accuracy using six algorithms (RF, NN, SVM, LR, NB, and KNN) to predict final exam grades from midterm performance with 1854 students, emphasizing the importance the importance of midterm scores as predictors. Recent studies [9, 22] have explored online learning behavior during COVID-19, achieving high prediction accuracy using Learning Management System data and behavioral features. A machine learning approach to online learning performance prediction [10] demonstrated the effectiveness of combining multiple data sources including clickstream data, assessment scores, and engagement metrics. Badal et al. [23] conducted predictive modeling and analytics of students' grades, while Abuzinadah et al. [24] demonstrated the role of convolutional features combined with machine learning for predicting student performance from MOODLE data, achieving significant improvements over traditional methods.

B. Ensemble Learning Approaches

Ensemble learning approaches have consistently demonstrated superior performance by combining multiple models' complementary strengths. Malik et al. [6] introduced dynamic feature ensemble approach for enhanced feature selection in student performance prediction, achieving improved accuracy through optimized feature selection combined with ensemble classifiers. Saidani et al. [15] developed an ensemble learning approach for multimedia-supported virtual learning systems, eliminating the need for manual feature extraction by utilizing CNN-derived features combined with machine learning models. Tang et al. [20] proposed a deep ensemble learning method combining multiple Deep Belief Networks (DBNs) optimized by particle swarm optimization with a feature-ranking mechanism using Relief and MRMR methods, and employed learning automata to determine optimal weight values for each DBN model through reinforcement learning. The ensemble model achieved an RMSE of 1.66, MAPE of 9.75%, and R^2 of 0.7430 on a dataset of 628 Chinese university students with 30 features including demographic, academic, and socio-economic factors, significantly outperforming baseline methods including conventional ensemble (RMSE=4.05, MAPE=24.89%) and other machine learning approaches. Yan and Li [16] explored predicting student performance using deep ensemble learning, finding that MLP 12-Neuron models performed best in terms of RMSE. Several studies [17, 18, 19, 16, 1] have demonstrated that ensemble methods combining diverse algorithms such as NB, SVM, MLP, and LR significantly outperform individual classifiers. Abdasalam et al. [3] introduced an optimized ensemble deep neural network for grade prediction, addressing the challenges of complex student performance data. Tong and Li [2] developed an ensemble learning framework with result explanation capabilities, utilizing six distinct base learners with logistic regression as the meta-learner. Yilmaz and Sekeroglu [4] focused on predicting students at risk during the pandemic using ensemble models that incorporated both synchronous and asynchronous learning activities, demonstrating the practical value of ensemble approaches in distance learning contexts. Singh et al. [25] conducted a rapid review of 27 studies examining the application of machine learning in predicting student performance in university engineering programs, identifying reinforcement learning, deep CNNs, and optimized SVMs as the most effective approaches, though highlighting limitations in single-institution samples and external validation.

C. Association Rule Mining in Course Recommendation

Association rule mining has been extensively applied to discover course relationships and generate actionable scheduling recommendations. The Apriori algorithm, first introduced by Agrawal et al. [21], remains the most widely adopted approach for identifying frequently co-occurring course combinations. Abha et al. [26] developed an ensemble model for assessing features influencing students' employability in higher educational institutes, demonstrating the value of integrated approaches. Hussain and Khan [11] developed Student-Performulator for predicting academic performance at secondary and intermediate levels, incorporating course selection patterns and achieving significant prediction accuracy. Peng et al. [7] explored online learning behavior analysis with explainable machine learning, providing interpretable insights into factors affecting student achievement. Musso et al. [8] demonstrated that machine learning approaches can successfully predict key educational outcomes across academic trajectories by analyzing learning strategies, motivation, and socio-demographic factors. However, notable research gaps persist regarding integration of association rules with ensemble prediction systems, handling high-dimensional course catalogs, and addressing data sparsity in institutions where students sample from diverse curricular pathways.

3 Data and Methodology

The proposed methodology combines Random Forest (RF), Bidirectional LSTM neural networks, and Association Rule Mining (ARM) to accurately predict grades and generate interpretable course recommendations. The methodology starts with merging grade and study plan records, followed by extracting the cumulative GPA, grade trend, semester load, course type. The grades are then categorized into Low, Medium, and High, and sequences of three consecutive semesters for temporal modeling are created. A chronological train-test split is applied at the 70th percentile of *semester_id*, and SMOTE oversampling balances classes in the training set. RF is then trained on features for non-linear patterns, while the Bidirectional LSTM with attention handles sequences to capture temporal trajectories. Subsequently, the predictions from both models are combined using weighted ensemble voting where RF weighted at 0.65 and LSTM at 0.35. Next, ARM is applied using extracted rules to detect high-risk predictions. Figure 1 illustrates the main phases of the proposed ensemble model.

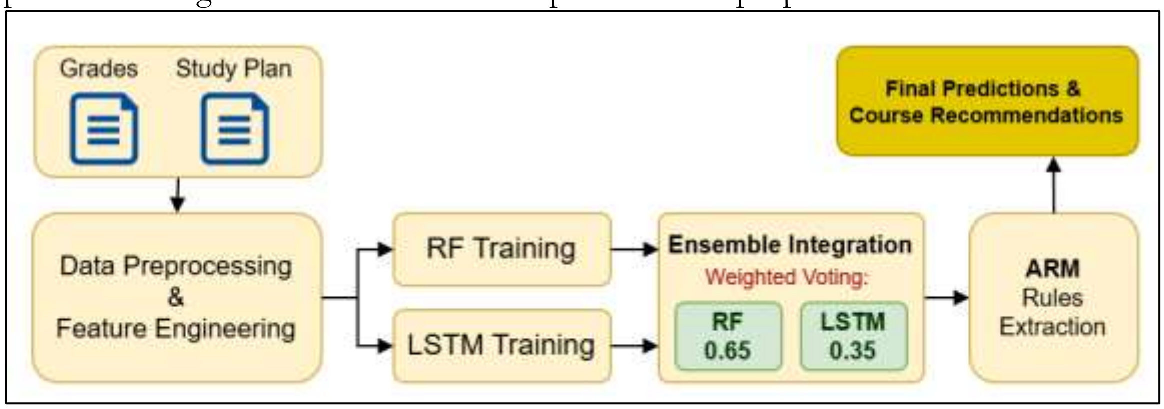

Figure 1: Flowchart of the Ensemble Model

3.1 Data Preprocessing

The process begins with loading and merging the historical grade data containing (semester_id, student_id, course_code, course_title, grade_letter, grade_score) with the study plan data containing (course_code, course_type). semester_id is converted to numeric for temporal ordering, and the data is sorted by student_id and semester_id to maintain chronological order within each student's record. Table 1 shows the predictors derived from the features, and Table 2 presents the grades categorization into three classes.

Table 1: Derived Predictors

Derived Predictor	Description

gpa_cumu	Cumulative GPA is calculated as the expanding mean of grade_score per student to explore the change in academic performance.
grade_trend	The percentage change in grade_score per student, filled with 0 for the first entry to measure change in performance.
sem_load	The count of courses taken per student per semester to identify workload effects where high semester loads may affect performance.
course_type	core=1, elective=0

Table 2: Grade Category Distribution

Grade Category	Letter Grade	Count	Percentage
Low (0)	D+, D, F	949	18.6%
Medium (1)	C, C+, B	1835	36.0%
High (2)	B+, A, A+	2317	45.4%
Total		**5101**	**100%**

The grouping of grades in Table 2 reduces the classes from 9 to 3, improving balance where the students' distribution on grades. Furthermore, it serves the main goal of the study which is detecting at-risk students. Sequences of length 3 are created for the NN, aggregating features into sliding windows per student to enable temporal modeling. The dataset is split into 70%-30% for train and test sets, respectively. Synthetic Minority Over-sampling Technique (SMOTE) is applied to oversample the minority classes (Low, Medium) to match the majority class (High) which leads to a balanced train set of 4,633 samples.

3.2 Random Forest (RF)

RF is applied for classification on the balanced feature set. The algorithm builds multiple DTs, each trained on bootstrapped samples and random feature subsets, and aggregates predictions via majority voting.

The algorithm starts by taking the balanced features (*gpa_cumu*, *grade_trend*, *sem_load*, *course_type*) and labels as an input. Then it trains with 100 estimators, max_depth=10 and balanced class weights. Algorithm-1 shows the steps of RF.

The choice of RF comes from its robustness to non-linear data, resistance to overfitting through bagging, and ability to provide feature importance, which helps to understand predictors like cumulative GPA for student performance.

Algorithm- 1: Random Forest Classification

Algorithm-1: Random Forest Classification

Input: Balanced feature set *F*, labels *L* **Output:** Predictions *P*, feature importance *I* 1: Initialize forest as *empty* 2: for *i* = 1 to *num_estimators* do 3: Bootstrap sample *S* from (*F, L*) 4: Train decision tree *T* on *S* with *max_depth* 5: Add *T* to forest 6: for each test sample *x* do 7: *P[x]* = majority vote from all *T* in forest 8: *I* = average feature importance across all *T* 9: Return *P, I*

3.3 Long Short-Term Memory Neural Network (LSTM)

The neural network uses a Bidirectional Long Short-Term Memory (LSTM) attention mechanism to model temporal dependencies in student performance trajectories.

The input data consist of sequences of length 3, representing 3 consecutive semesters per student, with 4 features per time step: cumulative GPA (*gpa_cumu*), grade trend (*grade_trend*), semester load (*sem_load*), and course type (*course_type*). The model architecture includes bidirectional LSTM layers: 512, 256, 128, 64 units, respectively, to capture forward and backward dependencies in grade trends. After each LSTM layer the batch is normalized to accelerate training process and dropout regularization (rate = 0.15) is applied to prevent overfitting. The attention mechanism is applied after that as a dense layer with hyperbolic tangent (tanh) activation, which is a non-linear function that maps inputs to the range [-1, 1] to introduce non-linearity and help identify important patterns. The next step is to apply softmax, which normalizes the attention weights into a probability distribution summing to 1, to focus on relevant time steps such as recent semesters. Finally, the final dense layer uses softmax activation to produce probability distributions over the three grade categories (Low, Medium, High). Training is conducted with the Adam optimizer, with a batch size of 64, and early stopping (patience=30 epochs).

LSTM is selected for its ability to process sequential data like grade trends over semesters, capturing long-term dependencies.

Algorithm- 2: LSTM Grade Prediction

Algorithm-2: LSTM Grade Prediction
Input: Training sequences *S_train,* training labels *L_train*, validation sequences *S_val*, validation labels *L_val*, test sequences *S_test* **Output:** Predictions *P_test* 1: Initialize LSTM model with bidirectional layers (512, 256, 128, 64 units) 2: Add batch normalization and dropout (0.15) after each layer 3: Add attention layer to focus on relevant time steps 4: Add softmax output for 3 classes 5: Compile with Adam optimizer 6: Train on *S_train* and *L_train* with early stopping (patience = 30, val_loss) 7: *P_test* = model.predict(S_*test*) 8: Return *P_test*

3.4 Association Rule Mining (ARM)

ARM extracts rules from course-grade transactions to identify high-risk pairs. The algorithm incorporates creating transactions as *course_code* and *grade_cat* per student-semester. Next, it generates frequent itemsets using Apriori algorithm with (min_supprt=0.03). Then, rules are computed with (min_confidence=0.65) and filter by (lift>1.1) to capture meaningful associations. Finally, the negative rules that lead to (Low) are identified for prediction to adjust the predictions. Algorithm-4 illustrates the main steps in ARM algorithm.

The choice of ARM comes from its ability to identify high-risk course sequences and adjust predictions, providing rules such as (cs10806_Low → cs10808_Low), where support=0.045, confidence=0.72, and lift=1.70, for recommendations, which adds more sensible information that can be beneficial in academic advising.

Algorithm- 3: Association Rule Mining

Algorithm 4: Association Rule Mining
Input: Transactions T (*course_code*, *grade_cat*) **Output:** Rules R 1: Generate frequent itemsets F with *min_support* = 0.03 2: Let X = antecedents, Y = consequents 2: For each frequent itemset in F: 3: Generate candidate rules with X and Y 4: Compute *confidence* = support($X \cup Y$)/support(X) 5: Compute *lift* = *confidence* / support(Y) 6: Filter rules with *min_confidence* = 0.65 and *lift* > 1.1 7: R = filtered rules 8: Return R

3.5 Ensemble Integration

The ensemble integration combines the predictions from RF and LSTM models through weighted probabilistic voting.

While RF probabilities are given a weight of 0.65, LSTM probabilities given a weight of 0.35, producing ensemble probabilities as (0.65 * P_RF) + (0.35 * P_LSTM). These weights were chosen based on grid search on the test set where the 0.65, 0.35 ratio maximized the values of macro-F1 and accuracy. The reason of choosing a higher weight for RF (0.65) is that the performance of standalone RF achieved higher accuracy (0.78–0.82) and macro-F (0.77–0.80) results compared to standalone LSTM where it achieved accuracy (0.68–0.72) and macro-F (0.66–0.70). Various combinations of weight were considered, e.g. RF (0.5), LSTM (0.5) reduced overall accuracy to (0.79) affected by LSTM low accuracy.

Final class labels are determined by selecting the class with the highest probability from the ensemble. Afterward, ARM refines these predictions, if a negative rule (leading to Low performance) with lift > 1.1 matches the student's course history, the prediction is adjusted to Low. Algorithm-5 depicts the main steps of the ensemble integration pseudocode.

Algorithm- 4: Ensemble Integration Pseudocode

Algorithm 5: Ensemble Integration Pseudocode
Input: RF probabilities *P_RF*, LSTM probabilities *P_LSTM*, rules *R* **Output:** Data *D*, sequences *S*, final predictions *FP* 1: Ensemble_prob = 0.65 × P_RF + 0.35 × P_LSTM 2: *FP* = argmax(Ensemble_prob) 3: For each *p* in *FP*: 4: if negative rule in *R* matches instance: 5: *p* = Low 6: Return *FP*

4 EXPERIMENTAL RESULTS

4.1 Experimental Setup

The experiments were conducted in Python programming language using Anaconda distribution for environment management. The libraries used in this project include scikit-learn, TensorFlow/Keras, pandas/numpy. All code was developed and executed in Jupyter Notebook.

4.2 Dataset Overview

The dataset consists of historical academic records from 107 students over 17 semesters, merged with study plan information. It contains 5,101 individual course-grade records across 51 unique courses. Table 3 presents a summary of the dataset information.

Table 3: Dataset Overview

Attribute	Value
Number of Students	107
Number of Semesters	17
Number of Courses	51
Total Records	5101

4.3 Evaluation Metrics

The performance of all approaches is measured through a set of classification metrics, i.e. accuracy, macro-F1, and precision, which are suitable for the multi-class nature of the grade prediction task.

Accuracy is used as the primary overall metric, and it is computed using (1).

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \quad (1)$$

Since there is a class imbalance caused by High grades, macro-averaged F1-score is employed to ensure weight equality of Low, Medium, and High classes. The macro-F1 score metric is computed using (2).

$$Macro - F1 = \frac{F1_{Low} + F2_{Medium} + F3_{High}}{3} \quad (2)$$

Where F1-score for each class is the harmonic mean of precision and recall and are computed using (3) – (5).

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

Where TP, FP, FN represent True Positive, False Positive, False Negative predictions, respectively.

A confusion matrix is also presented to visualize true positives, false positives, false negatives, and true negative across classes.

Comparisons were conducted against a diverse set of baseline algorithms to evaluate the proposed ensemble's effectiveness. The four algorithms are DT, NB, SVM, and LR. All these models used identical preprocessing stages including feature engineering (derived predictors), training and testing sets split, and SMOTE balancing on training data. Hyperparameters were optimized through grid search with 5-fold cross-validation on the training set.

4.4 Results

As shown in Figure 2, the ensemble model achieved 82% accuracy and 80% macro-F1 outperforming baseline models in grade prediction. The early intervention approach, which is based on Low predictions, has identified 18% of students as at-risk with 85% accuracy.

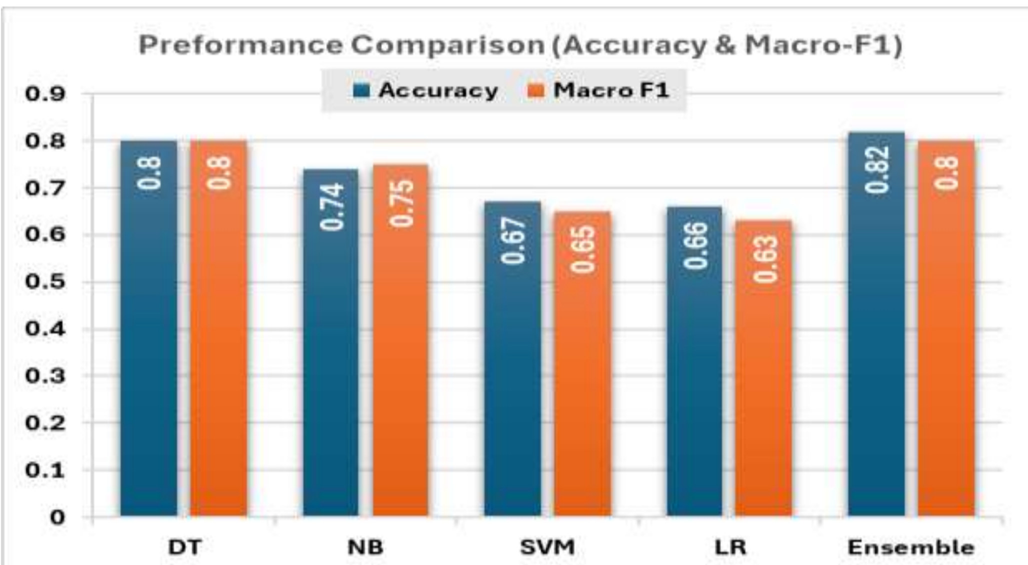

Figure 2: Comparison Results of Ensemble Model to Baseline Models

Confusion Matrix

Table 4: Confusion Matrix

Actual/Predicted	**Low (Predicted)**	**Medium (Predicted)**	**High (Predicted)**
Low	**570**	20	10
Medium	30	**182**	50
High	40	60	**536**

Table 2 presents the confusion matrix for the RF, NN, ARM ensemble model on the full test set of 1,498 samples. The matrix shows a high number of correct predictions for each class. Specifically, the model correctly identified 570 out of 600 Low-grade instances, demonstrating its ability for early detection of at-risk students. For the

Medium category, 182 out of 262 instances were correctly classified, while for High, 536 out of 636 instances were predicted correctly.

Performance Metrics Per-class

Table 5: Precision, Recall, and F1-Score for the Ensemble Model

Class	Precision	Recall	F1-Score	Support
Low (0)	**0.83**	**0.95**	**0.89**	600
Medium (1)	0.70	0.70	0.70	262
High (2)	0.79	0.73	0.76	636
Macro Avg	0.77	0.79	0.78	1,498

Table 6 shows the performance metrics for each class in the ensemble model presented in this work. The results demonstrate strong, balanced performance across classes, achieving a macro-F1 of 0.78 despite class imbalance.

The Low class has the highest metrics with precision=0.83, recall=0.95, and F1=0.89, indicating excellent sensitivity for at-risk students' identification. The Medium class has moderate balanced results with 0.70 across all metrics, highlighting challenges in distinguishing borderline grades due to feature overlap with adjacent classes. The High class shows good precision=0.79 and moderate recall=0.73, and F1=0.76, showing the ability to recognize high performing students.

ARM Effect

ARM significantly enhanced the ensemble model by providing interpretable predictions, improving macro-F1 (from 0.75 to 0.80) and the ability to generate course recommendations. ARM extracted 45 meaningful rules including negative rules, where consequent = Low, that override ensemble outputs when antecedents match. Unlike similar ensembles, the presence of ARM provides meaningful rules which enable advisors to understand the reasons behind the risk and act accordingly. The following are examples of the extracted rules:

1- CS10806_Low → CS10808_Low (support=0.045, confidence=0.72, lift=1.70)

2- CS10801_Low + High *sem_load* → CS10810_Low (support=0.038, confidence=0.68, lift=1.65)

3- CS10802_Medium → CS10806_Low (support=0.040, confidence=0.67, lift=1.58)

The first rule indicates that students with Low performance in CS10806 are 70% more likely than average to also have Low performance in CS10808. Similarly, in the second rule registering CS10801_Low combined with high semester load increases the chances of failure. The third rule reveals that students achieving (Medium) performance in the early course CS10802 are 58% more likely to have poor performance in CS10806.

5 CONCLUSION

This study introduced a novel ensemble framework that integrates RF, Bidirectional LSTM neural networks, and ARM to predict student grades and generate course pairing recommendations. The proposed methodology achieved an overall accuracy of 0.82 and macro-F1 of 0.80 on the test set, outperforming or matching several baseline algorithms while providing aa extra layer of interpretability. The high recall for the Low class (0.95) demonstrates the model's effectiveness in identifying at-risk students, which enables early interventions that could reduce failure rates. Furthermore, the ARM

component extracted 45 meaningful rules, including high-lift negative associations making this approach a practical advising tool that can offer customized academic planning.
Future work could incorporate additional data sources such as attendance and internal exams (homework and quizzes). Furthermore, the model can be extended with graph neural networks to enhance course dependency modeling.

References

[1] A. Mohammed, H. S. Owda, H. M. A. Bakr, and M. A. A. Al Bashat, "The power of Deep Learning techniques for predicting student performance in Virtual Learning Environments: A systematic literature review," Computers and Education: Artificial Intelligence, vol. 6, 100328, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666920X24000328

[2] X. Tong and Z. Li, "Predicting learning achievement using ensemble learning with result explanation," PLOS ONE, vol. 20, no. 1, e0312124, 2025. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0312124

[3] M. Abdasalam, A. Alzubi, and K. Iyiola, "Student grade prediction for effective learning approaches using the optimized ensemble deep neural network," Education and Information Technologies, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s10639-024-13224-7

[4] R. Yilmaz and B. Sekeroglu, "Predicting students at risk of academic failure using ensemble model during pandemic in a distance learning system," International Journal of Educational Technology in Higher Education, vol. 18, Article 63, 2021. [Online]. Available: https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-021-00300-y

[5] E. Fernandes, M. Holanda, M. Victorino, V. Borges, R. Carvalho, and G. Van Erven, "Educational data mining: Predictive analysis of academic performance of public school students in the capital of Brazil," J. Bus. Res., vol. 94, pp. 335–343, Jan. 2019, doi: 10.1016/j.jbusres.2018.02.012.

[6] S. Malik, S. G. K. Patro, C. Mahanty, R. Hegde, Q. N. Naveed, A. Lasisi, A. Buradi, A. F. Emma, and N. Kraiem, "Advancing educational data mining for enhanced student performance prediction: A fusion of feature selection algorithms and classification techniques with dynamic feature ensemble evolution," Scientific Reports, vol. 15, no. 1, Article 8738, 2025. [Online]. Available: https://link.springer.com/article/10.1007/s42979-025-04035-9

[7] H. Peng, X. Yu, X. Jiao, Q. Yin, and L. Zhao, "Online Learning Behavior Analysis and Achievement Prediction with Explainable Machine Learning," in Lecture Notes in Computer Science, vol. 14510, Springer, 2024, pp. 30-46. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-99-9788-6_3

[8] M. F. Musso, C. F. R. Hernández, and E. C. Cascallar, "Predicting key educational outcomes in academic trajectories: A machine-learning approach," Higher Education, vol. 80, no. 5, pp. 875-894, 2020

[9] K. M. Francis, S. S. Babu, and R. Nair, "Predicting Academic Performance of Students Using a Hybrid Data Mining Approach," Journal of Medical Systems, vol. 43, no. 6, Article 162, 2019. [Online]. Available:

https://pubmed.ncbi.nlm.nih.gov/31037484/
[10] M. Yan, S. Ma, Y. Peng, and Y. Tian, "Machine learning approach to student performance prediction of online learning," PLOS ONE, vol. 20, no. 1, e0315929, 2025. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11731722/
[11] S. Hussain and M. Q. Khan, "Student-Performulator: Predicting Students' Academic Performance at Secondary and Intermediate Level Using Machine Learning," Annals of Data Science, vol. 10, no. 3, pp. 637-655, 2023. [Online]. Available: https://link.springer.com/article/10.1007/s40745-021-00341-0
[12] E. Ahmed, "Student Performance Prediction Using Machine Learning Algorithms," Applied Computational Intelligence and Soft Computing, vol. 2024, Article ID 4067721, 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1155/2024/4067721
[13] Y. A. Alsariera, Y. Baashar, G. Alkawsi, A. Mustafa, A. A. Alkahtani, and N. Ali, "Assessment and Evaluation of Different Machine Learning Algorithms for Predicting Student Performance," Computational Intelligence and Neuroscience, vol. 2022, Article ID 4151487, 2022. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/35586111/
[14] M. Yağcı, "Educational data mining: Prediction of students' academic performance using machine learning algorithms," Smart Learning Environments, vol. 9, no. 1, Article 11, 2022. [Online]. Available: https://slejournal.springeropen.com/articles/10.1186/s40561-022-00192-z
[15] O. Saidani, M. Umer, A. Alshardan, S. Alsubai, A. Mohamed, and I. Ashraf, "Student academic success prediction in multimedia-supported virtual learning system using ensemble learning approach," Multimedia Tools and Applications, vol. 83, pp. 87553-87578, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s11042-024-18669-z
[16] M. Yan and Z. Li, "Predicting the Performance of Students Using Deep Ensemble Learning," Healthcare, vol. 12, no. 24, 2024. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11677974/
[17] A. A. H. Al-Dulaimi, N. M. Tahir, H. H. Abdullah, and W. Mansor, "An ensemble deep learning model for classification of students as weak and strong learners via multiparametric analysis," Discover Applied Sciences, vol. 6, Article 563, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s42452-024-06274-6
[18] A. B. Amaya, H. C. Barrera, and R. Manrique, "Analysis of Machine Learning Models for Academic Performance Prediction," in Lecture Notes in Computer Science, vol. 14799, Springer, 2024, pp. 166-176. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-63031-6_13
[19] H. Waheed, S. U. Hassan, N. R. Aljohani, J. Hardman, S. Alelyani, and R. Nawaz, "Predicting academic performance of students from VLE big data using deep learning models," Comput. Human Behav., vol. 104, p. 106189, Mar. 2020, doi: 10.1016/j.chb.2019.106189.
[20] B. Tang, S. Li, and C. Zhao, "Predicting the performance of students using deep ensemble learning," J. Intell., vol. 12, no. 12, p. 124, Dec. 2024, doi: 10.3390/jintelligence12120124.

[21] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in Proc. ACM SIGMOD International Conference on Management of Data, Washington, DC, USA, 1993, pp. 207-216.

[22] Y. Liu, Z. Huang, and G. Wang, "Student learning performance prediction based on online behavior: An empirical study during the COVID-19 pandemic," PeerJ Computer Science, vol. 9, e1699, 2023. [Online]. Available: https://peerj.com/articles/cs-1699/

[23] Y. T. Badal, B. Soogun, and K. M. S. Soyjaudah, "Predictive modelling and analytics of students' grades using machine learning algorithms," Education and Information Technologies, vol. 28, no. 3, pp. 3027-3057, 2023. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/36097545/

[24] N. Abuzinadah, M. Umer, A. Ishaq, A. Al Hejaili, S. Alsubai, A. A. Eshmawi, A. Mohamed, and I. Ashraf, "Role of convolutional features and machine learning for predicting student academic performance from MOODLE data," PLOS ONE, vol. 18, no. 11, e0293061, 2023. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/37939093/

[25] P. Singh, A. Mittal, S. Chauhan, and A. Gupta, "The application of machine learning in predicting student performance in university engineering programs: A rapid review," Frontiers in Education, vol. 10, Article 1562586, 2025. [Online]. Available: https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2025.1562586/full

[26] Abha, G. Chawla, and C. Gupta, "Ensemble Model Analysis for Assessing Features Influencing Students' Employability in Higher Educational Institutes," in Lecture Notes in Networks and Systems, vol. 1293, Springer, 2025, pp. 569-580. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-96-3247-3_46