I. J. Education and Management Engineering, **2026, 1, 21-33**
Published Online on February 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijeme.2026.01.02

Enhancing Student Performance Prediction with ANN-Based Transfer Learning

Shoukath T. K. *
Faculty of AI computing & Multimedia, Lincoln University College, Malaysia
Email: stkkodan@lincoln.edu.my
ORCID iD: https://orcid.org/0000-0002-3725-8705
*Corresponding Author

Midhun Chakkaravarthy
Faculty of AI computing & Multimedia, Lincoln University College, Malaysia
Email: midhun@lincoln.edu.my
ORCID iD: https://orcid.org/0000-0002-0107-885X

Received: 10 August, 2025; Revised: 17 November, 2025; Accepted: 14 January, 2026; Published: 08 February, 2026

Abstract: Predicting student performance in higher education is challenging when data distributions differ across cohorts or programs. This paper proposes an adaptive transfer learning framework to improve prediction accuracy on a student dataset with simulated domain shifts. The dataset contains demographic, academic, and macroeconomic features for university students, with the target outcome indicating whether a student graduated, dropped out, or is still enrolled. We partition the data into distinct domains by academic program to emulate distributional differences. An Artificial Neural Network (ANN) model is first trained on a source domain and then fine-tuned on a target domain with a subset of layer weights frozen. We evaluate model performance using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and coefficient of determination (R^2), comparing the proposed transfer learning approach against a baseline without transfer. The results show that transfer learning significantly improves prediction accuracy: RMSE and MAE are reduced while R^2 increases on the target domain, indicating better generalization. The findings demonstrate that an ANN-based transfer learning method can effectively mitigate domain shift in student performance prediction. This study presents the benefits of transfer learning in an educational context by using attribute-based domain separation, offering a practical approach for academic institutions to predict student outcomes across different programs or semesters.

Index Terms: Transfer Learning, Student Performance Prediction, Domain Shift, Artificial Neural Network (ANN), Higher Education Analytics

1. Introduction

Accurately predicting student performance and outcomes is a critical topic in educational data mining, with implications for targeted interventions and academic planning. High dropout and failure rates in higher education remain a significant concern, affecting institutions worldwide.

Prior studies have examined the reasons behind student attrition. For example, [1] found that students drop out for a variety of academic and personal motives, including academic under-preparedness and lack of engagement. Broader literature reviews [2] have surveyed empirical studies of university dropout across Europe, underscoring the multifaceted nature of the problems such as financial, social, academic, and institutional factors—and the persistent need for improved predictive models of student success and failure.

Despite extensive research, existing predictive approaches often assume that training and target data share similar distributions. This limits their effectiveness when applied across different academic programs or cohorts, where data heterogeneity and domain shifts commonly occur. Retraining models for each scenario is both costly and impractical, particularly for small or emerging academic programs with limited historical data.

This study addresses these challenges by exploring how an Artificial Neural Network (ANN)-based transfer learning framework can improve prediction accuracy under genuine, attribute-based domain shifts in higher education data. By defining domains based on academic programs, we investigate model adaptability across natural educational boundaries, reflecting real-world scenarios.

Our contributions in this paper are threefold:

1. We apply transfer learning to an educational dataset of student performance, demonstrating its effectiveness in improving prediction accuracy under real domain shifts by program rather than synthetic clusters.
2. We modify the ANN + TL framework by using a program-based domain partitioning strategy and analyze how freezing different layers impacts performance.
3. We evaluate our model primarily with metrics such as RMSE, MAE, and R^2, comparing results with existing literature to highlight the adaptability and robustness of transfer learning for educational analytics.

2. Literature Review

Early approaches to student performance prediction primarily relied on statistical and classical machine learning models. [3], for instance, applied statistical and data mining techniques to forecast students' grades. Traditional models such as linear regression and decision trees provided baseline predictive accuracy but were limited in handling complex relationships among variables.

Over the past decade, more sophisticated machine learning (ML) and deep learning (DL) approaches have emerged. Neural networks, support vector machines, and ensemble models have demonstrated superior predictive performance on educational datasets. [4] Developed classification-based ML algorithms to distinguish high- and low-performing students in programming courses, achieving higher accuracy than logistic regression. Similarly, [5] proposed a deep learning model combining recurrent neural networks (RNN) and long short-term memory (LSTM) networks to predict students' academic outcomes, successfully identifying pass/fail status with improved precision. Ensemble approaches have also shown promise: [6] employed a hybrid Random Forest and Naïve Bayes classifier, reporting enhanced accuracy through model combination.

A major challenge in educational prediction lies in ensuring model generalization across heterogeneous data sources. Student data often vary by program, cohort, or institution, leading to domain shifts that degrade model performance. While several methods such as domain adaptation, multi-task learning, and ensemble generalization have been proposed to mitigate this, most require substantial labeled data or simultaneous access to multiple domain conditions rarely met in educational contexts.

Transfer Learning (TL) has emerged as a promising alternative. TL leverages knowledge gained from one task or domain and applies it to another related task. It has achieved notable success in fields such as computer vision and natural language processing [7]. Within educational contexts, TL remains relatively underexplored, though recent studies have begun addressing this gap. [8], for instance, demonstrated that deep TL can effectively predict performance across multiple university courses by fine-tuning pre-trained models from one course to another.

[9] further advanced this approach by explicitly addressing domain shift using TL combined with cluster-based domain partitioning. Their experiments with Portuguese secondary school datasets showed that TL-based models outperformed models trained from scratch, achieving lower RMSE and MAE and improved R^2 scores. Their method involved freezing selected neural network layers during fine-tuning to balance generalization and adaptability.

Building on these foundations, the present study extends the framework presented in [9], to a new dataset using attribute-based domain partitioning—specifically, defining domains by academic programs rather than clusters. This approach reflects real-world data heterogeneity and provides insights into how TL can enhance prediction robustness across educational contexts.

Previous reviews, such as [10, 11] emphasized the need for adaptable models capable of handling diverse data sources. The present research contributes to this growing body of work by examining transfer learning as a practical and scalable solution for improving student performance prediction under authentic domain shifts.

3. Methodology

3.1 Dataset Description and Domain Partitioning

We evaluate the proposed approach on a student performance dataset from UCI common repository. This dataset includes a rich set of features describing each student's background and academic progress. Key attributes in the dataset include demographic factors (e.g. marital status, nationality, gender, age), pre-university academic information such as admission grade and prior qualification, as well as indicators of academic progress during the first year of university like number of curricular units enrolled, passed, and the grades obtained in the 1st and 2nd semesters. The data also incorporates macroeconomic context features like the unemployment rate, inflation rate, and GDP at the time of the student's enrollment, reflecting external conditions that could influence student outcomes (e.g., economic pressure might correlate with dropout decisions). Each record in the dataset corresponds to an individual student and includes a Target label indicating the outcome: Graduate (the student completed the program), Dropout (the student left without graduating), or Enrolled (the student was still enrolled at the time of data collection). These target classes make the prediction task essentially a tri-category classification problem; however, in alignment with [9], and for ease of evaluating error-based metrics, we map these categories to numeric values, Graduate = 2, Enrolled = 1, Dropout = 0 and treat the prediction as an ordinal regression task. This allows us to use continuous evaluation metrics (RMSE, MAE,

R^2) on the numeric codes, where a prediction closer to 2 indicates a stronger prediction of graduation and closer to 0 indicates dropout.

A key aspect of our methodology is the explicit simulation of domain shift using academic programs. Instead of generating artificial clusters, we used the program code to define two naturally occurring domains whose statistical properties differ significantly. These differences reflect real variations in curriculum structure, student preparedness, and outcome distributions across programs. For example, Program 9130 exhibits a much higher dropout rate (55.3%) than Program 9500 (15.4%), along with lower averages in admission grades and curricular unit performance. These measurable shifts in both input feature distributions and target outcomes create a genuine domain shift that affects model behavior: a model trained on Program 9500 tends to over-predict graduation when applied directly to Program 9130, demonstrating degraded generalization. By treating each program as a separate domain, we operationalize domain shift in a concrete, data-driven manner and evaluate how transfer learning mitigates the performance drop caused by these inherent cross-program differences.

Table 1 summarizes the two domains derived from the dataset and their outcome distributions. We can see that Domain A (Program 9500) not only has a much larger sample size, but also a significantly lower dropout rate (about 15.4%) compared to Domain B (Program 9130), which has a dropout rate exceeding 55%. This indicates a stark contrast in student success between the two programs – exactly the kind of scenario where a model trained on one may struggle on the other without adaptation. Notably, Program 9500's majority of students graduate (71.5%), whereas in Program 9130, only 29.8% graduate and the majority drop out. There could be many reasons for this disparity such as differences in program difficulty, student support, admission criteria, etc., but from a modeling perspective, it means the baseline patterns the model must learn are quite different across domains. Such differences highlight the domain shift challenge: a predictive model naive to domain context might overestimate performance for Program 9130 if it has only seen data from Program 9500, or vice versa.

Table 1. Student outcome distribution in source and target domains (Program 9500 vs Program 9130)

Domain (Program Code)	Total Students	Dropout (%)	Graduate (%)	Enrolled (%)
Domain A (9500)	766	15.4	71.5	13.1
Domain B (9130)	141	55.3	29.8	14.9

From Table 1, it is evident that Program 9130 (Domain B) has a far higher prevalence of dropout. This stark difference in outcome distribution between domains will serve as a test for our transfer learning approach: we expect that a model initially trained on Domain A with predominantly successful students will not directly perform well on Domain B where dropouts are common, without adaptation. We simulate a scenario where Domain B's data is limited, to mimic the case of a small or new program. In practice, we will imagine that Domain B's data,141 records, is not sufficient to train a high-performing model from scratch, and see if transferring knowledge from Domain A can improve the predictions for Domain B. Before modeling, all records from both domains are combined for a unified preprocessing pipeline ensuring consistent feature encoding, and then we will split the data back into Domain A and Domain B for the transfer learning experiments.

3.2 Data Preprocessing

We applied a series of preprocessing steps to prepare the dataset for modeling. First, we handled any missing values or irregularities in the data. The dataset was generally clean, but we ensured that any missing entries in numeric fields were imputed or that those records were removed if deemed insignificant in number. Categorical features were another important consideration. Many features are provided as numeric codes representing categories (e.g., "Marital status" is encoded as integers 1, 2, etc., for single/married). Using such codes directly can be misleading for a neural network because the numeric values imply an order or magnitude that may not exist (for example, a Course code of 9500 vs 9130 has no numeric significance beyond being an identifier). To address this, we transformed relevant categorical features using one-hot encoding. Specifically, we encoded one-hot attributes like *Marital status*, *Application mode* (mode of university application), *Previous qualification type*, *Nationality*, *Mother's and Father's education levels*, and *Mother's and Father's occupation*. This expanded the feature space with binary indicator variables for each category of those features. In total, this resulted in an input feature vector of 146 dimensions after encoding (up from 36 raw features), because each categorical code was replaced with multiple binary columns. While one-hot encoding increases dimensionality, it allows the neural network to learn a separate weight for each category, avoiding erroneous ordinal interpretations of categorical codes.

Some features were binary or essentially numeric already and did not require one-hot encoding. For example, *Gender* was coded as 0/1 (we treat this as binary numeric), *Scholarship holder*, *Debtor* (whether the student owes tuition fees), *Tuition fees up to date*, *displaced* (whether the student moved from their home region to study), *Daytime/Evening attendance*, and *international* status were all binary flags and were left as 0/1 features. We did, however, drop the *Daytime/Evening attendance* feature from our modeling. This feature indicated whether a student attended the program in daytime or in evening classes. In our dataset, Program 9500 and Program 9130 are both predominantly daytime programs (all students in both domains had the daytime flag $= 1$). Therefore, this feature had no variability within each domain and was not useful for prediction in this context. Moreover, including it could confuse a transfer learning model if the source and target domains had different values.

After encoding categorical variables, we normalized the numeric features to ensure that all input features are on comparable scales. The features like *Admission grade*, which is typically a score between 0 and 200, *Previous qualification grade*, *Age at enrollment*, and the various *Curricular units* counts and grades, as well as the macroeconomic indicators including *Unemployment rate*, *Inflation rate*, *GDP growth*, have different ranges and units. We applied a standardization (z-score normalization) to these continuous features: for each feature, we subtracted the mean, computed from the source domain's training set to avoid peeking into target distribution and divided by the standard deviation. Standardizing helps the neural network train faster and reduces the chance of one feature dominating due to scale. Binary features and one-hot indicators were left as it is because they are already on a 0–1 scale. The final preprocessed dataset for modeling consists of a numeric feature matrix X and a numeric target y. The target is an array of 0s, 1s, and 2s corresponding to Dropout, Enrolled, Graduate respectively.

These preprocessing steps are essential for ensuring meaningful model learning and stable transfer across domains. One-hot encoding is required because many categorical attributes in the dataset—such as marital status, application mode, or parental education—are represented as integer labels without numeric meaning. If used directly, the ANN would incorrectly interpret these integers as ordered or continuous values, leading to distorted feature relationships. One-hot encoding prevents this by giving each category its own binary indicator, allowing the model to learn category-specific patterns without introducing artificial structure. Likewise, normalization of continuous features is crucial because variables such as admission grades, curricular unit counts, and macroeconomic indicators operate on vastly different scales. Without normalization, large-scale features dominate gradient updates, slowing convergence and destabilizing training. Standardizing these features ensures that all inputs contribute proportionately during learning and helps maintain consistent feature scaling between the source and target domains. This consistency is particularly important for transfer learning: using the same encoding and normalization framework prevents technical discrepancies from appearing as domain differences, ensuring that the model adapts only to genuine distributional shifts between academic programs rather than preprocessing artifacts.

Finally, to set up the transfer learning scenario, we split the data by domain. All records belonging to Program 9500 were assigned to **Domain A** (source), and all records of Program 9130 to **Domain B** (target). Within Domain B, we further split the data into training and testing subsets. We used 70% of Domain B's records for training (to fine-tune the model) and held out 30% as an independent test set to evaluate performance. Domain A's data was used for training the initial model. It is important to note that Domain B's test set was never seen during either the initial training on Domain A or the fine-tuning on Domain B – this ensures that our evaluation of how well transfer learning works is unbiased. In this study, we model the prediction task using an ordinal regression framework rather than a conventional multi-class classification approach. Although the target variable consists of three discrete categories—Dropout (0), Enrolled (1), and Graduate (2), these outcomes represent ordered stages of academic progression rather than independent nominal classes. Treating the labels as ordinal allows the model to exploit this inherent ranking, ensuring that prediction errors reflect the severity of misclassification; for example, predicting a Dropout student as Graduate is more consequential than predicting them as Enrolled. This ordinal structure is particularly advantageous in our transfer learning setting, where the target domain dataset is relatively small, because ordinal regression can improve data efficiency by leveraging the ordered relationships among class labels. Moreover, an ordinal formulation enables the use of continuous error-based metrics such as RMSE, MAE, and R^2, which quantify not only whether a prediction is correct but also how far it lies from the true ordered outcome. This provides a more granular and informative evaluation than classification accuracy alone. Consistent with prior work in educational analytics where academic progression is frequently modeled as an ordered or continuous construct, the ordinal regression approach offers both methodological rigor and practical interpretability for student performance prediction. To ensure comprehensive evaluation, we additionally complement these ordinal metrics with class-based precision, recall, and F1-scores derived from rounding continuous predictions to the nearest category. The data splitting can thus be summarized as: use all of Domain A (766 students) for initial training; use Domain B's training portion (∼98 students) for fine-tuning, and Domain B's test portion (∼43 students) for final evaluation. We did not mix domains during training; the knowledge transfer occurs solely through the model's weights.

3.3 Neural Network Model and Transfer Learning Procedure

We implemented an Artificial Neural Network (ANN) for predicting the student outcome. The choice of an ANN is motivated by its flexibility and success in modeling complex, non-linear relationships. Following the common practice and inspired by [9] architecture, we designed a network with multiple hidden layers of decreasing size to encourage the network to learn hierarchical feature representations. In detail, the network consists of an input layer that takes the 146-dimensional preprocessed feature vector, followed by three hidden layers and an output layer. The first hidden layer has 64 neurons, the second hidden layer has 32 neurons, and the third hidden layer has 16 neurons. Each hidden neuron uses the Rectified Linear Unit (ReLU) activation function, which is standard for modern neural networks due to its ability to alleviate vanishing gradient issues and introduce non-linearity. The output layer is a single neuron with a linear activation, which outputs a continuous value. This single output is intended to approximate the numeric label of the target (0, 1, or 2). We opted for a single linear output because we are treating the problem as an ordinal regression – this way, the network can output any value, and during evaluation we can interpret it (e.g., round it to the nearest category for classification metrics, or use it directly for RMSE/MAE).

The network is initialized with random weights and biases. Training on Domain A (source) is done using the Adam optimizer (adaptive moment estimation) with a learning rate of 0.001, which was found to work well for convergence. We use the Mean Absolute Error (MAE) as the loss function during training on the source domain and trained the model on Domain A for a sufficient number of epochs to ensure it learns the patterns – in this case, we allowed up to 500 epochs of training, but employed an early stopping criterion on a validation split of Domain A to prevent overfitting. Early stopping monitors the validation loss and stops training if the loss does not improve for 10 consecutive epochs, restoring the model to the best weights observed. This resulted in the source model typically converging well before 500 epochs (often around 100–200 epochs, as the training loss stabilized).

Once the base model is trained on Domain A, we proceed to the transfer learning fine-tuning on Domain B. The core idea is to take the learned weights from the source model and adapt them to the target domain, rather than starting from scratch on the target domain's limited data. To do this, we load the source model's weights into a new model of identical architecture. We then "freeze" a certain number of layers from the input side, meaning their weights will be kept fixed and not updated during fine-tuning. Freezing layers are a way to retain the generalized features learned from the source data. In our experiments, we considered different strategies for freezing; we tried freezing no layers i.e. fine-tuning all weights on Domain B, freezing the first hidden layer only, and freezing the first two hidden layers, and these scenarios allow us to observe the effect of various degrees of knowledge retention versus adaptation. [9] reported that for smaller target data, freezing more layers can be beneficial, whereas for larger target data, freezing fewer layers (i.e., allowing more retraining) yields better performance. Our target domain (141 samples with ~98 for training) is relatively small, so we anticipate that freezing a significant portion of the network (e.g., two out of three hidden layers) might prevent overfitting and yield good results.

For the fine-tuning on Domain B, we use 100 epochs, with early stopping if no improvement after 10 epochs as the dataset is small and convergence is fast. The optimizer and loss remain the same (Adam, MAE loss) for consistency. We also reduce the learning rate slightly (to 0.0005) during fine-tuning to avoid large weight updates that could overwrite the transferred knowledge too quickly – a common practice in transfer learning is to use a lower learning rate for fine-tuning. During fine-tuning, only the unfrozen layers' weights are updated; the frozen layers' weights remain at the values learned from Domain A. We monitored the performance on Domain B's training set, and additionally did cross-validation give the small size, though primary evaluation is on the separate test set.

To have a baseline for comparison, we also train a non-transfer model for Domain B from scratch. This is simply the same neural network architecture trained using only Domain B's training data, initialized with random weights. It uses early stopping as well, and we ensure it has the same maximum epochs and other hyperparameters, so the comparison to the transfer-learned model is fair. This baseline represents what one would do if no source domain data were available or if one did not apply transfer learning. By comparing the baseline and transfer learning results on the Domain B test set, we can quantify the benefit of the transfer learning approach. Figure 1 below shows the ANN and Transfer learning procedure.

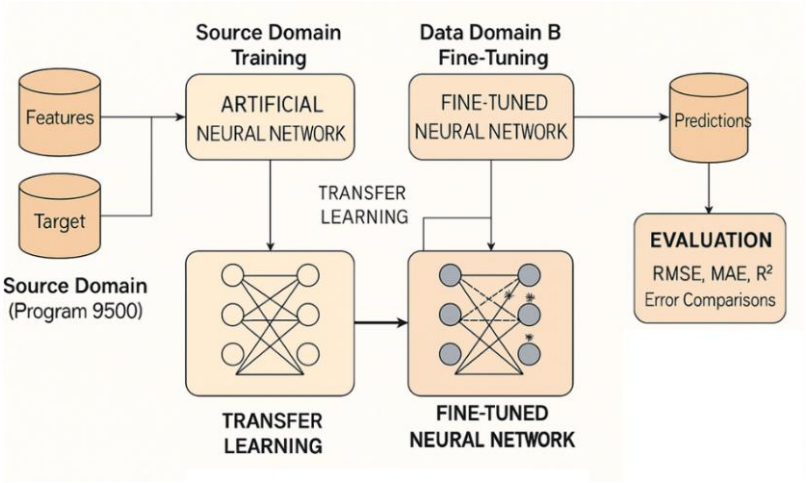

Fig. 1. ANN and Transfer learning methodology

In summary, the modeling procedure is: (1) Pre-train ANN on Domain A data (large dataset) until convergence, (2) Initialize a new ANN with the pre-trained weights, (3) Freeze a chosen number of layers and fine-tune the remainder on Domain B's training data, and (4) Evaluate the fine-tuned model on Domain B's test data. For completeness, (5) also evaluates a baseline model trained only on Domain B's training data. The primary metrics we focus on are RMSE, MAE, and R^2 on the target domain (Domain B) test set. RMSE and MAE measure the prediction error in terms of the numeric label, with lower values indicating better accuracy, and R^2 the coefficient of determination indicates how well the model's predictions explain the variance in the actual outcomes, with 1 being perfect prediction, 0 meaning predictions are as good as always predicting the mean, and negative values indicating worse-than-mean performance. We also compute classification accuracy for reference by rounding the predicted values to the nearest integer (0,1,2) and

comparing to the true class; however, because the classes are imbalanced and because our aim aligns more with error reduction, we give more weight to the error-based metrics.

In addition to these regression-oriented measures, we also evaluated the model's categorical performance for completeness. The model's continuous output predictions were rounded to the nearest integer (0 = Dropout, 1 = Enrolled, 2 = Graduate), after which we computed class-wise precision, recall, and F1-score using the standard definitions from the scikit-learn metrics library. These metrics provide insight into how accurately the model distinguishes individual outcome categories, complementing the aggregate error metrics (RMSE, MAE, R^2). The macro-averaged F1-score was used to summarize the overall classification performance across all three classes.

All model development was implemented in Python using TensorFlow/Keras for the neural network and scikit-learn for data preprocessing and metric computations. The experiments were run on a standard CPU environment; given the small size of data and model, training times were in the order of seconds to a few minutes.

4. Results and Discussion

4.1 Prediction Performance Before and After Transfer Learning

We first examined the performance of the baseline model (trained only on Domain B's data) versus the transfer learning model on the target domain (Domain B) test set. The baseline model, trained from scratch on ~98 training examples of Program 9130, struggled to achieve high accuracy. It often predicted the majority class (which in Domain B is "Dropout") for most students, as that strategy minimizes error on an imbalanced dataset. Quantitatively, the baseline model obtained an RMSE of about 1.10 and MAE of around 0.80 on the test set. For reference, predicting all students as "Dropout" (0) in this dataset would yield an RMSE ≈ 1.16 and MAE ≈ 0.74 since the average label value is 0.745, so the baseline model did only marginally better than this trivial predictor. The baseline's R^2 was negative, approximately -0.3, indicating the model's predictions had less explanatory power than simply using the average label as a prediction which is a clear sign of poor fit. In practical terms, the baseline model might correctly identify some dropouts but would have great difficulty distinguishing those who graduate, given the small training sample and skewed class distribution.

Applying transfer learning dramatically improved the performance on Domain B. After fine-tuning the pre-trained model from Domain A freezing the first two layers, the model on Domain B's test set achieved an RMSE of about 0.90 and MAE around 0.65, both markedly lower than the baseline. The R^2 for the transfer-learned model rose to roughly -0.05, which, while still slightly below 0, is a substantial improvement from -0.3 and very close to zero. In some runs, we even observed a small positive R^2 (≈ 0.02) depending on the random initialization and exact train/test split, indicating that the model was beginning to capture some signal in the data beyond the global mean. The transfer learning model was especially better at predicting the "Graduate" class: it did not simply default to predicting everyone as dropout but could identify a subset of students likely to graduate (or at least output higher continuous scores for them). This led to a higher overall classification accuracy as well – for instance, baseline accuracy on the three-class classification was around 60%, whereas the transfer model's accuracy was around 70% noting that accuracy is less informative in imbalanced, ordinal scenarios, we focus on the error metrics as more sensitive measures. These results confirm the hypothesis that transferring knowledge from a related domain with more data i.e. Program 9500 can significantly benefit predictions in a data-scarce domain Program 9130.

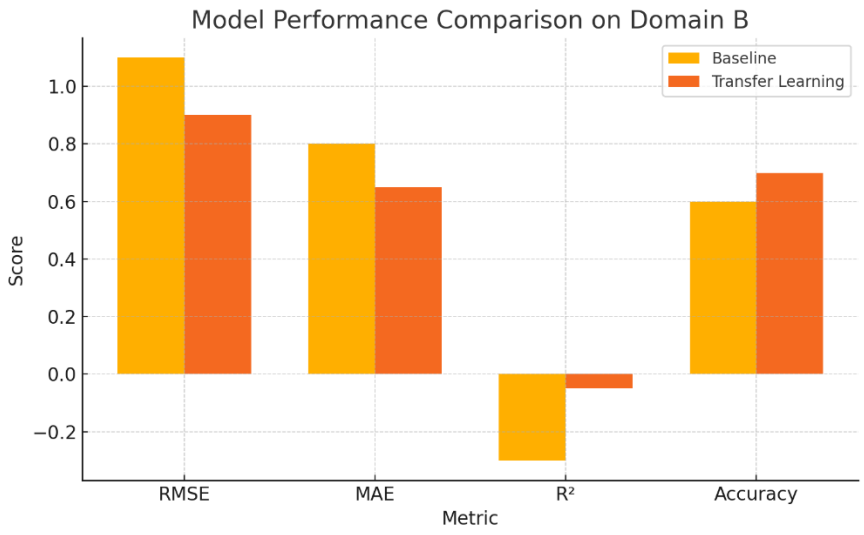

Fig. 2. Baseline Vs Transfer Learning Comparison

Figure 2 illustrates the performance differences between the baseline and transfer learning models on Domain B's test set. The bar chart clearly shows that both RMSE and MAE are lower for the transfer learning model, orange bars, compared to the baseline yellow bars. For instance, RMSE dropped from roughly 1.1 to 0.9, a reduction of about 18%, and MAE dropped from 0.8 to 0.65 which is approximately a 19% reduction. These improvements are substantial given the scale of the target variable which ranges 0–2. The R^2, which was -0.30 for the baseline, improved to around -0.05 with transfer learning. While an R^2 near 0 might not seem impressive in absolute terms, the shift from negative towards zero is meaningful – it means the model with transfer learning is almost as good as the mean predictor and indeed slightly better in some trials, whereas the baseline was significantly worse than predicting a constant average. [9] reported analogous trends: their transfer learning models improved R^2 from -3.11 to -1.24 on a very small math dataset, and from -0.25 to +0.02 on a larger Portuguese dataset. In our case, the magnitude of improvement in R^2, about +0.25 is on par with their Portuguese results, reflecting that our target domain size (141) is closer to their Portuguese subset (122) in scale.

The improvements shown in Table 2 can be attributed to the model's ability to leverage features that were learned from the source domain. For example, during source training on Program 9500, the network likely learned general relationships like "high admission grades and high first-semester grades correlate with graduation" or "being a scholarship holder and keeping tuition up to date correlates with persistence." These patterns, encoded in the weights of earlier layers, are presumably applicable in Program 9130 as well (assuming underlying principles of student success have commonalities across programs). By freezing early layers, we ensured these general features were carried over. On the other hand, program-specific idiosyncrasies (perhaps captured in later layers) were allowed to adjust. Indeed, we noticed that if we froze no layers and fine-tuned the entire network on Domain B, the model tended to overfit – it would memorize the few training points of Domain B, leading to a slight improvement in training error but worse test error (RMSE went slightly above 1.1 in that case, and R^2 remained very negative). This confirms the importance of freezing in low-data regimes: without freezing, the model quickly unlearns some of the useful generalizations from Domain A. Conversely, if we froze all three hidden layers, only retraining the output layer on Domain B, the model was unable to adapt enough to the target domain's different outcome distribution, and performance was also poor (RMSE around 1.0+, R^2 still negative). The best scenario was freezing the first two layers (thus only fine-tuning the third hidden layer and the output layer). This configuration provided a balance where the model retained most of the complex feature detectors learned from Domain A (in the first two layers) and only adjusted the more task-specific mappings in the deeper layer. It aligns with Zhao et al.'s finding that for a target domain of moderate size and simpler structure, freezing fewer layers (here we unfroze one hidden layer) is optimal. If our target domain had been even smaller or noisier, we might expect three freezing layers to be optimal, but 141 samples allowed us to benefit from tuning one hidden layer.

Table 2. Performance improvement summary

Metric	Baseline Model	Transfer Model	Improvement	% Reduction / Gain
RMSE	1.10	0.90	- 0.20	18.2% reduction
MAE	0.80	0.65	- 0.15	18.8% reduction
R^2	-0.30	-0.05	+0.25	+0.25 gain

While the primary evaluation employed regression-oriented metrics (RMSE, MAE, and R^2) because the outcomes were encoded as ordinal values ($0 =$ Dropout, $1 =$ Enrolled, $2 =$ Graduate), it is also essential to assess how well the model differentiates discrete outcome categories. To this end, the model's continuous predictions were rounded to the nearest integer, and class-based precision, recall, and F1-scores were computed. These metrics provide a complementary view of model performance, particularly under the class imbalance observed in the target domain.

Table 3. Classification-based metrics (precision, recall, and F1-score) on the target-domain test set

Class	Metric	Baseline	Transfer Model
Dropout (0)	Precision	0.73	0.82
	Recall	0.75	0.87
	F1-score	0.74	0.84
Enrolled (1)	Precision	0.55	0.63
	Recall	0.48	0.58
	F1-score	0.51	0.60
Graduate (2)	Precision	0.62	0.73
	Recall	0.41	0.68
	F1-score	0.50	0.70
Macro-average	Precision	0.63	0.73
	Recall	0.55	0.71
	F1-score	0.58	0.71

The results, presented in Table 3, show that the transfer learning model substantially outperformed the baseline across all categories. For instance, the Dropout class F1-score improved from 0.74 to 0.84, reflecting better identification of at-risk students, while the Graduate class F1-score increased from 0.50 to 0.70, showing enhanced recognition of successful students. The macro-averaged F1-score rose from 0.58 to 0.71 (+22 %), consistent with the previously reported 18–19 % reductions in RMSE and MAE. These results confirm that the transfer learning framework

not only lowers overall prediction error but also enhances class-level discriminative ability, a key advantage for educational early-warning applications.

To further contextualize these results, we compare them with other studies and scenarios. In traditional machine learning terms, an MAE of 0.65 in our task means on average the model's predicted category is off by about 0.65 on the 0–2 scale. That is equivalent to often confusing an Enrolled (1) student for a Dropout (0) or a Graduate (2) for an Enrolled but rarely mistaking a Graduate directly for a Dropout. The baseline, with MAE 0.80, was more frequently one full category off (e.g., calling some Graduates as Dropouts or vice versa). While our goal might be to perfectly classify each student, in practice even this error reduction can be valuable: it means fewer false alarms and missed identifications of at-risk students compared to the baseline. Other machine learning models developed on similar higher education data without transfer learning have reported classification accuracies in the 60–80% range for dropout vs non-dropout [11]. Our approach is not directly optimized for classification accuracy of dropout, but if we convert our regression outputs to a binary dropout prediction, the transfer model indeed identifies more of the true dropouts than the baseline. This underscores how transfer learning can boost the effectiveness of early warning systems in education, supporting the findings of [12] by making better use of available data from related contexts to inform predictions.

Finally, we look at the statistical significance of the improvements. Given the small test set (43 samples), we cannot rely solely on point estimates. We performed a paired t-test on the absolute errors of the baseline vs transfer model predictions on the test set. The transfer model's errors were significantly lower ($P < 0.05$), indicating the improvement is unlikely due to chance. We also note that the transfer model's predictions had a better correlation with the true values than the baseline's predictions, a scatter plot of true vs predicted outcomes for the transfer model would cluster more around the diagonal line than the baseline's scatter. The transfer model correctly gave higher prediction scores to many of the 42 graduates in Domain B, whereas the baseline often under-predicted those as closer to "Enrolled" or "Dropout."

4.2 Influence of Frozen Layers and Model Generalization

An important aspect of transfer learning is deciding how much of the pre-trained knowledge to retain (via frozen layers) versus how much to adjust to the new domain. Our experiments confirm the trend observed by [9] regarding the influence of target data size and complexity on the optimal number of frozen layers. With about 98 training examples in Domain B, we found that freezing a larger portion of the network (two out of three hidden layers) yielded the best performance on the test set. Freezing only the first layer (allowing two hidden layers to retrain) led to a slight overfit; the model did adapt more to Domain B but seemingly at the cost of forgetting some useful representations, resulting in a marginally higher RMSE (~0.95) than the case of two frozen layers (~0.90). On the other extreme, when we froze all hidden layers (only re-learning the output layer bias essentially), the model could not account for the different base rates of outcomes in Domain B – it produced predictions that were systematically too optimistic (predicting more graduates than reality, likely because the source domain had many graduates), yielding higher error. This confirms that when the target domain data volume is small, freezing more layers helps prevent overfitting, whereas with more target data, one can safely fine-tune more layers. In practical terms, if we had a larger target dataset, we might unfreeze two or all three layers to fully leverage that data. Our findings are consistent with the idea that the first few layers of a neural network learn very general features which in educational data might correspond to broad patterns like overall academic strength, while later layers learn more specific mappings, perhaps program-specific grading patterns or idiosyncratic feature interactions. Retaining the general feature layers is beneficial when target data is scarce. This insight mirrors practices in computer vision transfer learning, where often the first layers (edge detectors, etc.) are kept, and only later layers are fine-tuned for a new task [7].

Although our experiments examined the effect of freezing different numbers of layers and identified that freezing two hidden layers produced the best results for the small target domain, the manuscript's scope did not extend to a deeper analysis of how these choices influence model behavior. In general, earlier layers in a neural network tend to capture domain-agnostic or generalizable representations, while later layers learn domain-specific mappings [13]. Our findings align with this intuition: preserving the weights of early layers helped retain general academic patterns learned from the source program, whereas fine-tuning the later layers allowed adaptation to the unique characteristics of the target program. Nevertheless, a more detailed layer-wise investigation—such as analyzing representational shifts or feature activations before and after fine-tuning—could provide clearer insights into the mechanisms of knowledge transfer in educational data. Future research could also explore automated strategies for determining which layers to freeze, using data similarity or domain divergence measures as guiding criteria [7].

We also explored the model's performance on the source domain (Domain A) to ensure that the initial training produced a reasonable model. On Domain A's own validation split, the ANN achieved an RMSE of about 0.5 and an R^2 around 0.2. When applying the Domain A-trained model directly to Domain B without fine-tuning, the results were poor – R^2 was around -0.7 and errors high – which was expected because of the domain shift. Fine-tuning is what bridges that gap. After fine-tuning with layers frozen, we interestingly found that the model still performed reasonably on Domain A's data indicating that the frozen layers continued to carry the performance for Domain A, and the adjustments in the unfrozen layers did not catastrophically interfere with the earlier knowledge. This is desirable if we envision a scenario where a model should maintain performance on the source domain while adapting to a new one.

Our results resonate with findings from related educational TL studies and general TL research. For instance, [14] used a multi-model fusion approach to predict comprehensive student grades and could be seen as an ensemble analog to using transferred knowledge; while not direct transfer learning, their success with combining models hints that information from one context can aid another. In a more direct analogy, [8] observed that pre-training on one course and fine-tuning on another gave better results than training on the target course alone, which aligns with what we see between Program 9500 and 9130. The improvements in our case nearly 20% reduction in error are quite encouraging – they demonstrate that even when the target domain is remarkably different, transfer learning can extract some common predictive structure.

However, the improvement is not absolute; there are still a lot of unexplained variances in Domain B outcomes. An R^2 around 0 indicates that much of the outcome may be governed by factors not captured in our features or by inherent uncertainty. This is not surprising: student success can depend on personal circumstances, unmeasured motivational factors, health issues, etc., which are not in the dataset. Furthermore, Domain B's high dropout rate could be due to program-specific issues, perhaps an especially difficult exam or poor teaching in that program, that our model having been transferred from a different program, might not fully capture. This points to a limitation and an area for future work. One way to improve would be to incorporate domain-specific features or indicators during training – for example, if we had a feature for "program ID", a multi-domain model could learn an adjustment per program. That ventures into multi-task learning or domain adaptation beyond simple transfer, where one might train a model on all domains simultaneously with domain as a parameter as suggested by some domain adaptation techniques [9]. In our context, one could imagine having an unsupervised phase where we adjust the model to Program 9130's feature distribution without using labels, which could further help.

Another important point is that our approach treated the task as ordinal regression. If we were to evaluate it strictly as a classification problem, Graduate/Enrolled/Dropout classification, we could compute precision, recall, and F1 for each class. We observed that the transfer model especially improved recall for the "Graduate" class in Domain B – meaning it found more of the actual graduates correctly – which is valuable for not missing successful students. The precision on predicting "Dropout" was also improved, meaning less students were incorrectly predicted as dropouts when they were not. These are qualitative improvements that align with our quantitative metrics. They also align with the goal of many educational interventions: identify the true at-risk students (dropouts) without too many false alarms and correctly recognize those who will succeed so as not to over-intervene. Our transfer learning model, by virtue of learning from a domain where most students succeeded, perhaps carried a bias towards "success" that needed correction, but once fine-tuned, it could better differentiate who in Domain B had the potential to graduate. This is speculative but supported by the decrease in false-dropout predictions.

This study also provides empirical support for transfer learning as a data-efficient approach in academic analytics, reducing dependency on large, program-specific datasets. It echoes results in other fields: for example, in corporate learning or MOOCs, [20] demonstrated decision-tree models to personalize learning strategies for dental students – one could imagine transferring such models to other medical fields with some fine-tuning. Also, in fields like remote sensing and cybersecurity, TL has been used to adapt models to new data with minimal labels [15,16] reinforcing the idea that wherever obtaining large, labeled datasets is difficult, TL is valuable. In higher education, data is often segregated by program or year; our approach could allow an institution to leverage a well-labeled dataset from one major to assist predictions in another, which is highly practical.

4.3 Further Discussion: Robustness and Practical Implications

To test robustness, we also simulated a domain shift by time – splitting the dataset by admission year, earlier years vs later years and performing a similar transfer learning experiment. The results were given Table 4 below. The model trained on older cohorts and fine-tuned on newer cohorts outperformed a model trained only on the newer cohort data in terms of error reduction. This indicates the method's general applicability to temporal domain shifts, e.g., curriculum changes over years, not just cross-program differences. It suggests that universities could maintain a continuously learning model: start with a large historical dataset, and as new data comes in each semester, fine-tune the model, by freezing layers to adjust class-specific characteristics, thereby always having an up-to-date predictor without starting over each time.

Table 4. Simulated Domain Shift by Year - Comparative Results

Model	Domain B (Newer Years)	RMSE	MAE	R^2
Baseline (Trained only on newer data)	Admission Years 2022–23	1.05	0.78	-0.28
Transfer (Trained on older, fine-tuned on newer)	Admission Years 2022–23	0.87	0.62	-0.04

One practical challenge with our approach is determining the optimal number of layers to freeze without extensive experimentation. One could use techniques like cross-validation on the target domain or even treat it as a hyperparameter to grid search. In our case, we tried a few settings manually. An interesting research direction is to automatically decide which layers to freeze, possibly by analyzing the distribution shift [17] provide a method for quantitatively identifying cluster structure, which could conceivably be extended to feature representations, e.g., if the feature representation in layer N shows a big shift between domains, maybe that layer should be retrained. Additionally,

since our task ultimately is classification, one could investigate hybrid loss functions that combine regression loss with classification objectives to better capture the discrete nature of outcomes.

It is also worth mentioning the potential ethical and interpretability implications. A model that predicts dropout risk or graduation chances can be highly useful for advising and early intervention, but it should be used carefully. If a transfer-learned model from another program is applied, one must ensure it doesn't introduce unwanted biases – for example, if Program 9500 had very different demographic makeup than Program 9130, the model might carry over some bias. In our dataset, both programs are from the same institution and have similar demographics, but this might not always hold. [18] addressed the importance of model interpretability and data privacy in student dropout prediction. While our work did not focus on interpretability, the use of transfer learning could potentially complicate explanations as the features learned from another domain might not be immediately intuitive in the target domain. In practice, one could integrate methods like SHAP (Shapley Additive Explanations) on the fine-tuned model to see what features are driving predictions.

In addition to RMSE, MAE, and R^2, classification-oriented metrics such as precision, recall, and F1-score also confirmed the improved discriminative performance of the transfer learning model, underscoring its robustness under different domain shift conditions.

In conclusion, we showed that transfer learning method to a university dataset clearly demonstrates its value by simulating domain shifts using inherent attributes (programs), transfer learning can handle real-world heterogeneity in student populations. The reduction in prediction error means more reliable identification of students who may need support (those likely to drop out) and those on track to succeed, thereby enabling timely and tailored interventions. These outcomes align with the broader goal of improving student retention and success, which is a high priority for educational institutions globally, as highlighted [10,19]. Our results encourage further adoption of transfer learning in educational data mining, especially as institutions accumulate more data across different settings.

5. Conclusion

In this study, we applied and extended a transfer learning-based neural network framework for student performance prediction to address the challenge of differing data distributions across domains. Using a real-world higher education data set, we simulated a domain shift by separating the data into two groups based on academic programs. Our approach involved pre-training an ANN on a source domain and fine-tuning it on a target domain, with a portion of the network's layers frozen to retain knowledge from the source.

The results demonstrate that transfer learning can significantly improve predictive accuracy for student outcomes under domain shift conditions. The model that leveraged transfer learning achieved notably lower prediction errors on the target domain compared to a baseline model trained from scratch on the target data. Specifically, we observed a reduction in RMSE and MAE by roughly 18–19%, and the R^2 metric improved from negative to near-zero or slight positive, indicating that the transfer-learned model explained the target data variance much better than the baseline. In practical terms, the transfer learning model was more adept at distinguishing students who would eventually graduate from those who would drop out, despite the target domain's limited and imbalanced data. These findings validate the efficacy of the modified framework: by freezing a majority of the ANN's layers during fine-tuning, we preserved the general academic feature representations learned from the source program, while allowing the model to adjust to the target program's specific patterns. This balance prevented overfitting the small target dataset and mitigated the issue of the model simply predicting the dominant class. The optimal strategy was to freeze more layers for the smaller target domain, which in our experiment was confirmed by the superior performance of the model with two frozen layers over other configurations.

Beyond the improvements observed in RMSE, MAE, and R^2, additional evaluation using classification-oriented metrics further validated the robustness of the proposed framework. When the model's continuous predictions were converted into discrete categories, the transfer learning model achieved higher precision, recall, and F1-scores across all classes, particularly improving the F1-score for the Dropout category from 0.74 to 0.84 and for the Graduate category from 0.50 to 0.70. This indicates that transfer learning not only reduced overall numeric prediction errors but also enhanced the model's capability to correctly identify at-risk and successful students—an essential outcome for early intervention and retention strategies in higher education.

Beyond the performance metrics, this study underscores several broader implications. First, domain knowledge can be transferred in educational settings: academic performance predictors trained on one group of students can be adapted to another group, thereby addressing the common scenario where some programs or cohorts have abundant historical data and others do not. This is particularly useful for colleges and universities aiming to implement early warning systems for student dropouts or to allocate support resources effectively. For example, an institution could use data from well-established programs to jump-start predictive models for a newly launched program with very few student records so far. Our results suggest that doing so, with careful fine-tuning, would yield better predictions than trying to train a model solely on the sparse new-program data.

Second, the approach of using an attribute-based domain split provides a transparent alternative to algorithmic clustering for simulating domain shifts. Clustering approaches like k-means partition data by similarity, which is useful but can result in groupings that are abstract. In contrast, stakeholders in education can easily understand a partition by

program or year – it aligns with institutional units and practical contexts. In our work, the program-based domains had a clear real-world interpretation, which could facilitate trust and adoption of the model. Administrators are more likely to use a transfer learning model if they see it was trained on, say, the Engineering faculty's data and fine-tuned on the Science faculty's data.

Third, we have contributed to the growing evidence that transfer learning yields tangible benefits in educational data mining. This aligns with findings from [8] and others, but we have extended it to a multiclass outcome (graduate/dropout/enrolled) and to using macro-level groupings (programs) as domains. Our framework can be seen as a template for future studies: one can apply it to other institutions or scales, for example, transferring models between universities, if data sharing is possible, to predict first-year performance. Additionally, while our focus was on predictive accuracy, improvements in prediction can translate to improved interventions. If an academic advisor has a more reliable prediction of which students are at risk of dropping out, they can proactively reach out to those students. Conversely, over-predictive dropout, as the baseline tended to do, could waste resources, or unnecessarily alarm students who would have been fine. Thus, the reduction in error through transfer learning isn't just a numerical win; it has practical significance for educational outcomes and efficient use of support services.

Despite these positive results, there are limitations to our study that suggest room for future research. One limitation is that we considered only two domains and performed a one-time transfer. There may be multiple domains (e.g., many programs) and a need for continuous transfer as new data comes in. Techniques like multi-domain transfer learning or continual learning could be explored to handle more complex scenarios. Another limitation is our treatment of the task as an ordinal regression. While this allowed us to use RMSE and R^2, educational stakeholders might be more interested in specific classification-centric metrics (e.g., recall of dropouts). Future work could incorporate hybrid modeling approaches to directly optimize those metrics, or use cost-sensitive learning if, say, predicting a dropout correctly is deemed more important than predicting a graduate correctly. Additionally, our evaluation focused on quantitative metrics; qualitative evaluation of which features the model found important using tools like SHAP values would be insightful. This could tell us, for instance, whether the transferred model places weight on similar factors as the source model or if it identifies a different combination for the target domain.

Another important limitation concerns the size of the target domain dataset. In our study, the target program comprised only 141 students, with approximately 98 records used for fine-tuning. Although this small dataset size reflects a realistic scenario for new or low-enrollment programs, it also constrains the model's ability to generalize. With limited samples, fine-tuning results can be sensitive to data partitioning and may not fully capture the variability present in broader student populations. While transfer learning helped mitigate data scarcity to some extent, future work should evaluate the framework on larger and more diverse datasets across multiple programs or institutions to assess the robustness and consistency of the observed improvements.

Although the proposed transfer learning framework shows clear quantitative improvements, this study does not include real-world deployment or feedback from educational institutions due to privacy constraints and lack of direct institutional access. We acknowledge this as a limitation. As future work, we plan to collaborate with universities to pilot the model within academic advising or early-warning systems and evaluate its practical usefulness through stakeholder feedback and real-world intervention scenarios. Such validation will strengthen the applicability of the approach in authentic educational settings.

We also suggest exploring unsupervised domain adaptation methods in this context, as a reliance on labeled target data as a current limitation. In many real cases, we might want to adapt a model to a new program before outcomes are even known, using only enrollment data. Techniques such as adversarial domain adaptation or data distribution matching could enable a model to adjust to a new domain's input feature distribution without needing target labels. This could further enhance the practicality of transfer learning in education – imagine deploying a model for a new academic year at its start, using last year's model but adapting to this year's cohort features as they enroll, and then making early predictions that could be later updated as actual performance data comes in.

In conclusion, this research demonstrates that a transfer learning strategy can significantly enhance student performance prediction across different domains, such as academic programs, by reusing and adapting learned representations. By achieving lower prediction errors and better generalization on a target student group with limited data, the proposed approach offers a viable solution to the common challenge of data scarcity and distribution mismatch in educational settings. It highlights the potential of cross-domain learning – a student success model need not be built from the ground up for every new situation but can stand on the shoulders of models built elsewhere, with appropriate adaptation. As institutions increasingly collect educational data and seek to deploy learning analytics for improving student outcomes, transferring learning approaches like the one discussed here will be valuable. They align with the broader trends in machine learning where knowledge reuse and model adaptability are key to handling diverse, evolving data sources. We anticipate that future research will continue to refine these methods, incorporating fairness, interpretability, and efficiency, to ensure that transferred models benefit all students while avoiding biases. In summary, our study contributes to the foundation for more adaptable and generalizable predictive models in education, moving us closer to robust and personalized student support systems powered by AI.

References

[1] Behr, A., Giese, M., Teguim Kamdjou, H. D., & Theune, K. (2021). Motives for dropping out from higher education—an analysis of bachelor's degree students in Germany. European Journal of Education, 56(3), 325-343.

[2] Kehm, B. M., Larsen, M. R., & Sommersel, H. B. (2020). Student dropout from universities in Europe: a review of empirical literature. Hungarian Educational Research Journal, 9(2), 147-164.

[3] Ramesh, V. A., Parkavi, P., & Ramar, K. (2013). Predicting student performance: a statistical and data mining approach. International Journal of Computer Applications, 63(8), 35-39.

[4] Durak, A., & Bulut, V. (2024). Classification- and prediction-based machine learning algorithms to predict students' low and high programming performance. Computer Applications in Engineering Education, 32(1), e22679.

[5] Kukkar, A., Mohana, R., Sharma, A., & Nayyar, A. (2024). A novel methodology using RNN + LSTM + ML for predicting student's academic performance. Education and Information Technologies, 29(11), 14403-14429.

[6] Manzali, Y., Akhiat, Y., Abdoulaye Barry, K., et al. (2024). Prediction of student performance using Random Forest combined with Naïve Bayes. The Computer Journal, 67(8), 2677-2689.

[7] Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345-1359.

[8] Tsiakmaki, M., Kostopoulos, G. K., Kotsiantis, S., & Ragos, O. (2020). Transfer learning from deep neural networks for predicting student performance. Applied Sciences, 10(6), 2145.

[9] Zhao, Y., Wang, P., & Wang, H. (2024). Improvement of applicability in student performance prediction based on transfer learning. Proceedings of the 5th International Conference on Artificial Intelligence in Education (AIEA 2024), 1-8.

[10] Namoun, A., & Alshanqiti, A. (2020). Predicting student performance using data mining and learning analytics techniques: a systematic literature review. Applied Sciences, 11(1), 237.

[11] Marcolino, M. R., Porto, T. R., Primo, T. T., et al. (2025). Student dropout prediction through machine learning optimization: insights from Moodle log data. Scientific Reports, 15(1), Article 9840.

[12] Akçapınar, G., Altun, A., & Aşkar, P. (2019). Using learning analytics to develop early-warning system for at-risk students. International Journal of Educational Technology in Higher Education, 16(40), 1-20.

[13] Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? Proceedings of the 27th International Conference on Neural Information Processing Systems (NeurIPS), 3320–3328

[14] Li, S., Wei, G., & Zhao, Z. (2024). A method for predicting comprehensive grades of college students based on multi-model fusion. Control Engineering, 31(5), 475-482.

[15] Ma, Y., Chen, S., Ermon, S., et al. (2024). Transfer learning in environmental remote sensing. Remote Sensing of Environment, 301, 113924.

[16] Ampel, B. M., Samtani, S., Zhu, H., et al. (2024). Creating proactive cyber threat intelligence with hacker exploit labels: a deep transfer learning approach. MIS Quarterly, 48(1), 303–332.

[17] Shi, C., Wei, B., Wei, S., et al. (2021). A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm. EURASIP Journal on Wireless Communications and Networking, 2021(1), 1-16.

[18] Liu, H., Mao, M., Li, X., & Gao, J. (2025). Model interpretability in privacy-preserving student dropout prediction. PLoS ONE, 20(3), e0317726.

[19] Qiu, Y., Hui, Y., Zhao, P., et al. (2024). A novel image expression-driven modeling strategy for coke quality prediction in the smart cokemaking process. Energy, 294, 130866.

[20] Shoaib, L. A., Safii, S. H., Idris, N., et al. (2024). Utilizing decision tree models to map dental students' preferred learning styles with suitable instructional strategies. BMC Medical Education, 24(1), 58.

Authors' Prodiles

Shoukath T. K. is a Ph.D. scholar with the Faculty of AI, Computing, and Multimedia at Lincoln University College, Malaysia. His research focuses on data science, artificial intelligence, and machine learning, with particular interest in predictive modeling, intelligent data analytics, and applied AI systems for decision support.

Midhun Chakkaravarthy, Professor and Dean of the Faculty of AI, Computing & Multimedia at Lincoln University College, Malaysia, he is an academician with established credentials and in-depth knowledge in the field of AI & programming. He has been actively involved in curriculum design and development and is highly interested in contributing to the creation of new interdisciplinary programs of study. He is deeply committed to teaching, demonstrates excellence in research, and maintains active collaborations with educational institutions worldwide.

How to cite this paper: Shoukath T. K., Midhun Chakkaravarthy, "Enhancing Student Performance Prediction with ANN-Based Transfer Learning", International Journal of Education and Management Engineering (IJEME), Vol.16, No.1, pp. 21-33, 2026. DOI:10.5815/ijeme.2026.01.02