TabNet–XGBoost Hybrid Model for Student Performance Prediction and Customized Feedback

Anupama Prasanth
College of Computer Studies, University of Technology Bahrain, Salmabad, Kingdom of Bahrain

***Abstract*—Virtual Learning Environments (VLEs) have emerged as a cornerstone of modern education, enabling large-scale delivery of learning materials, assessments, and interactions in fully or partially online formats. The dynamic and self-paced nature of VLEs makes the early prediction of learner scores crucial for timely intervention and support. The existing frameworks either underperform in capturing complex, non-linear relationships in heterogeneous educational data or lack interpretability mechanisms necessary for actionable interventions. This study proposes a TabNet–XGBoost hybrid model with SHAP-based interpretability for score range classification in VLE contexts, using the Open University Learning Analytics Dataset (OULAD). Data preprocessing involved cleaning, encoding, normalization, feature engineering, and score band derivation, producing an enriched feature matrix integrating demographic, assessment, and engagement indicators. TabNet's sequential attentive feature selection extracted a latent representation of the most informative variables, which was subsequently refined by XGBoost to produce sharper decision boundaries for four distinct score ranges. SHAP values were computed post-prediction to identify domain-specific performance drivers, enabling alignment with a structured feedback module across seven predefined learning domains. Experimental results demonstrated a classification accuracy of 98.8% on the test set, outperforming the baseline frameworks. The SHAP-driven feedback mechanism provided interpretable, domain-targeted insights, enhancing the model's practical applicability for educators and academic support teams. By integrating high predictive accuracy with transparent reasoning and actionable feedback, the proposed framework addresses both the technical and pedagogical requirements of early performance prediction in online learning environments, offering a scalable solution for real-time academic monitoring and intervention.**

Keywords*—*Virtual learning environments; student performance prediction; TabNet; XGBoost; SHAP; feedback generation; quality education

I. Introduction

Virtual Learning Environments have transformed the educational landscape by enabling interactive, resource-rich, and quality education platforms where learners can access course materials, engage with peers, and receive instructional guidance regardless of physical location [1]. These digital ecosystems offer unprecedented flexibility, as students can customize learning pace and learning paths and integrate academic activities into diverse personal and professional schedules [2]. VLEs provide scalability for institutions, support diverse learning modalities, and facilitate continuous assessment opportunities. Despite these advantages, learner performance within such environments can vary significantly, influenced by factors such as digital engagement habits, self-directed learning skills, and adaptability to technology-mediated instruction [3]. A central challenge remains the consistent and accurate evaluation of learner progress, alongside the timely provision of constructive feedback [4]. Without proactive performance monitoring and targeted intervention, learners in VLEs may experience reduced motivation, disengagement, or even withdrawal from the course.

Traditional approaches for performance assessment in VLEs often rely on periodic quizzes, assignment submissions, and final examinations. While these methods are administratively convenient and familiar to both educators and learners, they are predominantly retrospective, identifying performance gaps only after significant time has elapsed. Such delay limits the scope for early corrective action and diminishes the effectiveness of feedback, particularly in fast-paced or highly competitive academic contexts [5]. Moreover, these approaches tend to prioritize academic scores over other formative indicators such as participation in discussions, interaction with learning resources, or the quality of peer collaboration, all of which are vital for a holistic understanding of learner progress in virtual environments.

The adoption of advanced analytics in VLEs, with machine learning (ML) and deep learning (DL) algorithms, predictive analytics, and learning dashboards, enables educators to track and forecast student performance using a wide array of behavioral and academic indicators [6]. However, existing implementations often fall short in adaptability, failing to dynamically adjust to evolving learner behaviors over time [7]. Some models focus narrowly on quantitative academic metrics, neglecting qualitative engagement data, while others operate as opaque "black boxes", limiting their interpretability and diminishing trust among educators and stakeholders. This underscores the need for an integrated, transparent, and adaptive performance prediction framework tailored for VLEs. An ideal system should deliver early, accurate forecasts of learner performance, informed by both academic and behavioral data, while ensuring that feedback is specific, actionable, and timely.

The primary objective and questions addressed in this study are mentioned below:

- Development of a hybrid TabNet–XGBoost Framework that integrates TabNet's attention-driven feature selection with XGBoost's gradient-boosted decision refinement, optimized for score range classification in VLEs.

- Implementation of an interpretable feedback generation module that maps SHAP-based feature importance values to seven predefined learning domains, enabling personalized, domain-specific recommendations for academic performance improvement.

The succeeding sections of the study are organized as follows: Section II offers a comprehensive analysis of the latest advances in student performance prediction in VLEs and highlights the current research constraints. Section III delineates the proposed methodology. The experimental results are highlighted in Section IV along with a comprehensive assessment of the model's functionality. Finally, Section V completes the research by summarizing the main conclusions and highlighting the possible domains for future research.

II. Related Work

Liu et al. [8] suggested the use of temporal engagement data collected from online learning platforms for student performance prediction. The OULAD, consisting of clickstream data from 5,341 distance learners across 12 different VLEs, was the source of the study. Two primary feature sets were designed based on the weekly and monthly aggregation of click counts, treating them as structured panel data. A number of classifiers, including Logistic Regression (LR), k-Nearest Neighbors (KNN), Random Forest (RF), Gradient Boosted Trees, 1D Convolutional Neural Networks (1D-CNN) and Long Short-Term Memory networks (LSTM) were compared in the study. LSTM outperformed others with a maximum accuracy of 90.25% and interactions with the homepage, subpages, content, and quizzes were identified as the most predictive behaviors in the feature importance analysis. As the framework failed to incorporate assessment performance or generate personalized feedback, its applicability in student score improvement was limited.

Arashpour et al. [9] proposed hybridized ML models for student exam performance prediction utilizing OULAD. The hybrid models integrated Support Vector Machines (SVM) and Artificial Neural Networks (ANN) with a Teaching–Learning-Based Optimizer (TLBO) for both classification and regression. TLBO performed feature selection and optimized the ANN structure in parallel, identifying the optimal subset of predictive input variables. The SVM model, with eight selected features, achieved a classification accuracy of 86.10%, while the ANN model achieved 84.94% on the test data. For regression, Support Vector Regression (SVR) outperformed ANN, achieving a correlation coefficient $R > 0.7$. Engagement, measured through clickstream behavior and performance in ongoing assessments, was identified as the most influential predictor. Heavy dependence on numerical engagement metrics, lacking semantic interpretation or personalized feedback mechanisms, hampered the practical use of the framework.

Rao et al. [10] proposed DL models for early prediction of student academic performance utilizing log data obtained from academic systems. The dataset comprised log records from 108 students enrolled in an "Information Science" course was utilized which captured sequential interactions with the learning environment. The architecture employed a Recurrent Neural Network (RNN) that processed sequential student interaction logs, capturing temporal dependencies in learning behavior patterns over the course timeline. The framework dynamically updated the internal state with each input sequence, enabling it to learn correlations between past engagement and future academic outcomes. The RNN framework with 84.3% accuracy outperformed the Decision Tree classifier and ensemble learning approaches. As there was no integration of time-based engagement trends or automated feedback generation, the deployment of the framework for continuous formative assessment and personalized academic support was constrained.

Jawad et al. [11] proposed a Random Forest classifier, enhanced by the SMOTE oversampling technique, to predict student academic performance in VLE. The study utilized OULAD and constructed dynamic student profiles across six-time intervals (120 to 260 days), integrating engagement patterns and assessment scores. The framework was retrained per time segment to simulate real-time academic monitoring, and the integration of SMOTE mitigated the impact of imbalanced class distributions. The model achieved its highest testing accuracy of 84.2% at 260 days, suggesting that richer late-term data boosts predictive strength. Additional benchmarking revealed that XGBoost and Logistic Regression achieved accuracies of 84.3% and 80.9%, respectively. The reliance on separate models across multiple time intervals introduced structural redundancy and limited scalability for real-time deployment.

Yu et al. [12] proposed a recurrent neural network (RNN) for student learning outcome prediction in an online Artificial Intelligence course, focusing on identifying at-risk students using a limited number of commonly available LMS features. A number of DL models: Simple Recurrent Network (SRN), LSTM, Gated Recurrent Unit (GRU), Multilayer Perceptron (MLP), and CNN, as well as ML frameworks: LR, SVM, RF, and decision trees (DT), were compared. The SRN and CNN models achieved the highest overall accuracy (93%), while LSTM and GRU followed closely with 91% and 92%, respectively. Further week-wise performance analysis revealed that GRU showed the best predictive accuracy (87%) by the 18th week, followed by SRN (86%) and CNN (84%), outperforming all classical ML models. The rigid framework for all prediction weeks overlooked varying learning dynamics at different stages of the course, hampering adaptability to fluctuating student engagement patterns.

Kusumawardani and Alfarozi [13] proposed a transformer encoder-based DL framework for early prediction of student performance using log activity data from learning management systems (LMS). The study utilized the OULAD and was designed for both daily and weekly prediction tasks for the early identification of at-risk students. The transformer architecture was fine-tuned by evaluating the effects of components of positional encoding, feature aggregation strategies and weighted loss functions. The framework performed best without positional encoding, and weekly feature aggregation yielded higher accuracy. For the withdrawn vs. pass-distinction task, an accuracy of 83.17% was observed at just 20% of the course duration, improving further to 90% accuracy by the end. Across all tasks, the transformer outperformed LSTM by 1-3% in accuracy and 3-7% in F1-score. The model complexity and resource-intensive architecture hindered deployment in real-time or low-resource educational environments.

Chen et al. [14] proposed Explainable Student Performance Prediction (ESPP) for early detection of at-risk learners in VLEs. The data for the study was collected from a Digital Transformation course at Gadjah Mada University, Indonesia, comprising weekly student activity logs during online instruction. The hybrid deep learning framework integrated CNN and LSTM networks, alongside the hybrid SMOTE oversampling technique to balance class distributions. CNN layers extracted spatial patterns from temporal activity sequences, which were then passed through LSTM layers to model time-dependent engagement trends. Conv-LSTM model achieved the highest accuracy of 91%, followed by CNN-LSTM at 88%, outperforming LSTM (85%), SVM (80%) and LR (64%) on evaluation in the six-week. However, the computational complexity was a major drawback, hindering scalability or real-time deployment in low-resource educational platforms.

Adnan et al. [15] suggested a predictive model to identify at-risk students in VLEs, MOOCs, and LMS platforms using OULAD. A number of multiple ML and DL algorithms were evaluated, including RF, DT, SVM, LR, KNN, MLP, ANN, and Naive Bayes (NB). The study incorporated key variables such as assessment scores, clickstream data indicative of engagement intensity, and temporal indicators [15]. On evaluation, the RF classifier performed consistently across different stages of course progression and achieved 79% accuracy at 20% course completion, which further improved to 88% at 60% and peaked at 91% at 100% of the course length. The high dependence on feature engineering that did not generalize well across different online learning environments without significant adaptation constrained the potential of the study.

Riestra-González et al. [16] proposed a course-agnostic framework to predict student performance from LMS log files in the early stages of course delivery. The study utilized data from 5,112 courses hosted on Moodle at the University of Oviedo, involving over 29,000 students across diverse disciplines. Multiple classification models were employed, such as DT, NB, LR, SVM, and MLP, to detect at-risk, failing, and exceptional students at different course phases. The MLP model achieved the highest accuracy, ranging from 80.1% at 10% of course length to 90.1% at 50%, while DT followed closely with accuracies ranging from 79.5% to 89.6% over the same intervals. A clustering technique was also applied to identify six consistent student interaction patterns, four of which were identified to be strongly correlated with student performance: early answering of quizzes, prompt viewing of LMS resources, early viewing of course assignments, and procrastination in viewing course content, which indicated a higher risk of failure. The reliance on static interaction patterns, without incorporating the temporal sequence or evolution of student behaviors over time, hampered the generalizability of the framework.

Yağcı [17] proposed a machine learning-based predictive framework for predicting students' final exam grades using minimal yet impactful academic indicators. The dataset for the study comprised midterm grades, faculty affiliation, and departmental information of 1,854 undergraduate students enrolled in the Turkish Language-I course at a public university in Turkey. Six ML models: RF, SVM, LR, NB, KNN, and Neural Network (NN) were evaluated in the study. The framework extracted predictive insights from numerical academic features without incorporating behavioral or interaction data. On evaluation, RF and NN achieved the highest classification accuracy, both reaching 74.6%, followed by SVM with 73.5%, LR at 71.7%, NB at 71.3% and kNN at 69.9%. The sole focus on academic variables without the incorporation of behavioral or engagement-related features limited the study.

Chen et al. [18] proposed an Attention-Based ANN (Attn-ANN) model for early prediction of at-risk students by integrating attention mechanisms across both time and feature dimensions. Data for the study were collected from the M2B system at Kyushu University, which included LMS-based records such as attendance, report submissions, and course access. The framework employed dual attention layers: on the temporal dimension to identify important weeks and on the feature dimension to prioritize learning activities. The attention weights are integrated directly into a standard artificial neural network, enabling it to dynamically adjust the influence of each time step and feature during the learning process. The Attn-ANN achieved an accuracy of 64.3% in the Programming Techniques (PT) course and up to 89.5% in the Digital Signal Processing (DSP) course, outperforming conventional models like MLP, LSTM, and GRU in both early and progressive weeks. The sensitivity of attention weight calibration required manual adjustment or retraining across different course structures and educational settings.

Hakkal and Ait Lahcen [19] suggested integration of XGBoost with logistic regression-based models for learner performance prediction within Intelligent Tutoring Systems (ITS). Three regression models were utilized: Item Response Theory (IRT), Performance Factor Analysis (PFA), and DAS3H. Eight real-world datasets from varied sources, including four ASSISTments skill-builder math datasets, two KDD Cup Algebra datasets, the Statics engineering dataset, and a new Moodle-Morocco dataset, were employed in the study. The framework analyzed historical ITS interaction logs and estimated the probability of a learner answering future questions correctly. The XGBoost-enhanced PFA outperformed standard PFA in seven datasets, DAS3H also improved on the ASSISTments17 dataset, while IRT's performance remained stable across datasets, indicating less benefit from XGBoost integration. The highest accuracy value of 84.9% was observed for DAS3H-LR on Bridge-Algebra06, followed by 84.3% by PFA-XGBOOST on the Bridge-Algebra06 dataset. The worst accuracy of 68.1% was exhibited by IRT-XGBOOST on the Assistments09 dataset. However, XGBoost consumed a long execution time for large datasets and required advanced hyperparameter tuning and specialized feature encoding, hampering the scalability in real-world scenarios.

Iatrellis et al. [20] proposed a two-phase machine learning approach combining unsupervised and supervised learning in student outcome prediction. Data from the Computer Science Department at the University of Thessaly, Greece, was utilized in the study to forecast degree completion time and the likelihood of student enrollment. The first phase employed the K-Means algorithm to cluster students based on educational factors and metrics, which identified three coherent student groups. The prediction models were developed for each cluster in the second phase for customized predictions. The clustering-

guided model outperformed non-clustering models, achieving an accuracy of 80.5%, compared to 75.5% in the non-clustering approach. As the student grouping was solely from a data-driven perspective, the faculty insights or academic advisor evaluations were overlooked in the study, hampering the content relevance of the predicted outcomes.

A. Research Gap

Despite significant advancements in predictive modeling for student performance, several research gaps persist. Many existing approaches, including DL architectures like BiLSTM, CNN, and LSTM, as well as ML models such as RF, SVM, and hybrid optimization-based methods, have demonstrated high accuracy using datasets such as OULAD, LMS logs, or Intelligent Tutoring Systems [8] [14] [22]. However, a recurring limitation is the narrow scope of input features; most frameworks rely heavily on clickstream data, assessment scores, or basic academic records, overlooking rich contextual, socio-cultural, and behavioral attributes that could improve model generalization [15] [17]. Furthermore, temporal modeling has often been static or rigid, failing to adapt to evolving engagement patterns across course stages [12]. While certain methods incorporate early prediction capabilities, few integrate automated personalized feedback or intervention strategies to translate predictions into actionable academic support [10]. Scalability is also a challenge; resource-intensive architectures like transformers and CNN-LSTM struggle in low-resource educational settings, while approaches with segmented time-based retraining introduce redundancy. Lastly, class imbalance handling techniques like SMOTE improve accuracy but may reduce real-world applicability where such balancing is infeasible, highlighting the need for models that maintain robustness in naturally imbalanced datasets [11]. This calls for a novel model that could predict the student performance in VLEs with high accuracy and tailor feedback for academic improvements while considering both academic and behavioral attributes.

III. Materials and Methods

The proposed study integrates TabNet and XGBoost for enhanced student performance prediction in VLEs. The hybrid architecture leverages TabNet's sequential decision-step feature selection and representation learning capabilities, coupled with XGBoost's robust non-linear modeling, to generate precise score range classifications. The OULAD serves as the primary data source and SHAP was employed for the customized feedback generation. Fig. 1 represents the basic architecture of the proposed model.

A. Dataset Description

The proposed study employs the publicly available Open University Learning Analytics Dataset (OULAD) from Kaggle, a comprehensive real-world dataset sourced from the VLE of the Open University, the largest distance-learning institution in the United Kingdom [21]. The dataset comprises records from seven distinct modules delivered across multiple academic years, with presentations denoted for the first and second semesters. It integrates three major data categories: student demographic attributes, assessment-related data, and VLE interaction logs. The student demographic attributes include age band, gender, geographic location, disability status, prior education level, and Index of Multiple Deprivation (IMD) band.

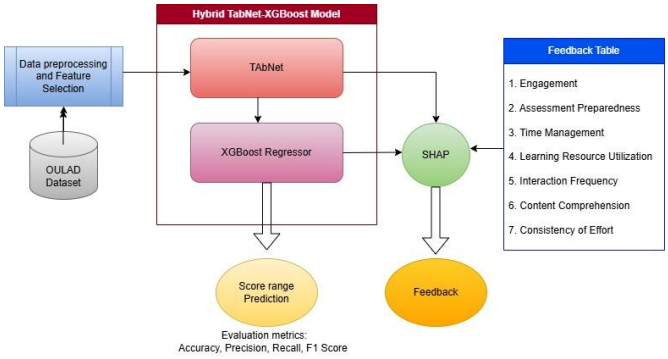

Fig. 1. Basic architecture of the proposed model.

The assessment-related data contains marks, submission dates, and completion status for assignments, quizzes, and final examinations. The VLE interaction logs include time-stamped clickstream data capturing student engagement patterns, including content access, forum participation, and quiz activity. The dataset's multimodal nature, combining demographic, behavioral, and performance indicators, aligns closely with the proposed model architecture, enabling both sequential learning through historical interactions and explainability through feature-level attribution. Given its richness, scale, and diversity, OULAD provides an ideal foundation for developing high-accuracy predictive models with actionable, domain-specific feedback to improve student performance outcomes

B. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is performed for the systematic examination of the structural composition and statistical properties of the dataset, enabling the identification of underlying patterns, distributions, and relationships between variables. The process is crucial for the detection of anomalies, missing values, and potential data imbalances, while also highlighting key behavioral and demographic trends relevant to student performance. Insights from the EDA informed both engineering decisions and the subsequent design of the predictive modelling framework.

The distribution of results across individual modules illustrated in Fig. 2, reveals substantial variation in student performance patterns, offering critical insights for targeted academic interventions.

As depicted, certain modules, such as BBB and FFF (as named in the dataset), exhibit the highest overall enrolments and are characterized by a comparatively large proportion of withdrawn students, suggesting potential structural or delivery-related challenges. Conversely, modules like AAA and EEE display lower participation volumes but notable failure rates, which may indicate concentrated difficulties among smaller cohorts. The presence of distinctions is relatively modest across all modules, with only minor variation between courses, while pass rates remain the most frequent positive outcome. This module-level stratification provides a more nuanced understanding than aggregate performance statistics, enabling the identification of courses where tailored pedagogical

strategies or enhanced learning support may yield the most significant improvements. Such granularity aligns directly with the proposed model's capacity to generate module-specific performance predictions and customized feedback, thereby facilitating informed decision-making for educators and administrators aiming to reduce attrition and improve academic success rates.

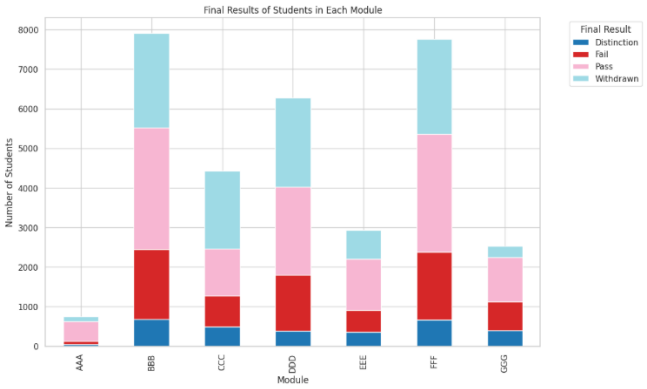

Fig. 2. Final results of students in each module.

Fig. 3 represents the distribution of key learning activity features, segmented by students' final results.

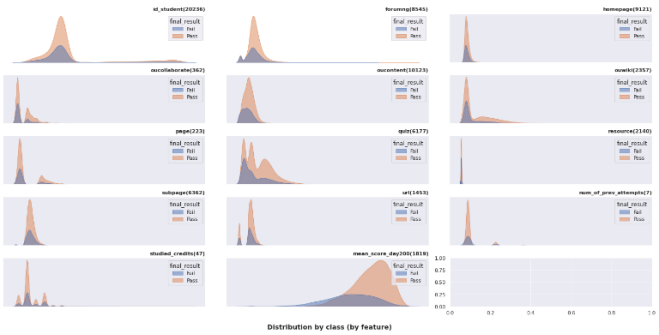

Fig. 3. Distribution of key learning activity features.

Each subplot corresponds to a specific feature from the OULAD dataset with kernel density estimations illustrating the relative frequency of values for each outcome class. Across most activity metrics, students who passed tend to show higher engagement levels, as indicated by a broader spread and higher density in the upper value ranges compared to their failing counterparts. Notably, features like quiz and mean score day demonstrate a clear separation between the two classes, indicating strong predictive value. Conversely, some variables, such as oucollaborate and ouwiki, exhibit overlapping distributions, implying limited discriminative power. The skewed patterns in the number of previous attempts and studied credits suggest that prior academic history and course load may influence performance outcomes.

Fig. 4 compares the average submission dates between B Semesters and J Semesters (for the second and first semesters, respectively) expressed in days from the start of the academic term. The results show that on average, students in the second semesters submit their work earlier compared to those in the first semesters. This difference of nearly 10 days suggests possible variations in course scheduling, assessment deadlines or student pacing between the two semester types. Such temporal patterns can inform predictive modeling by highlighting semester-specific behaviors that may influence engagement and performance outcomes.

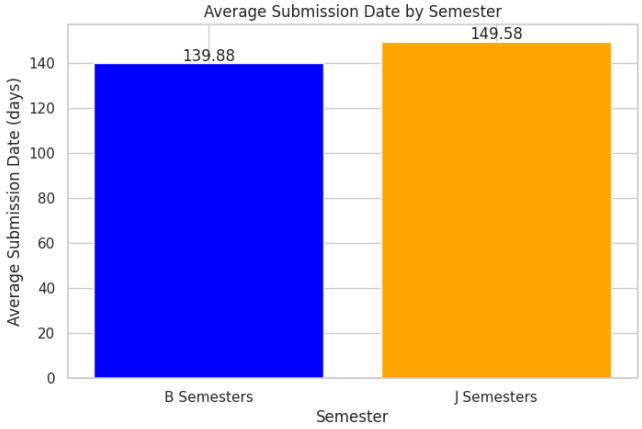

Fig. 4. Average submission date by semester.

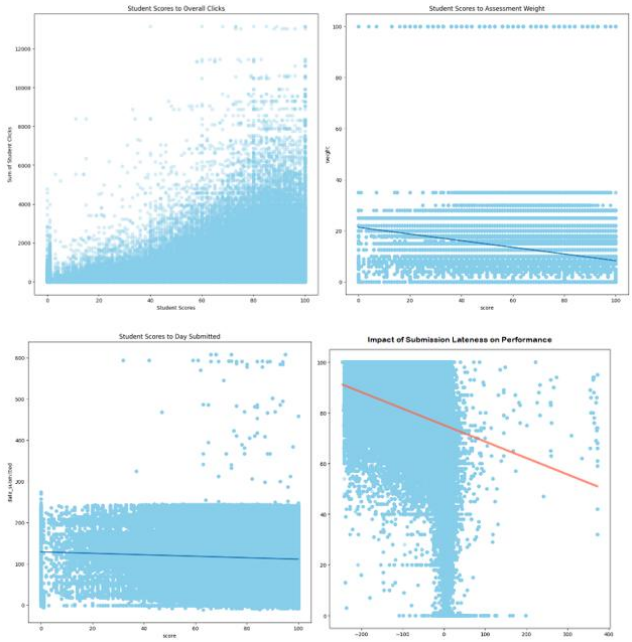

Fig. 5. Student score analysis.

Fig. 5 presents scatter plot analyses exploring the relationships between student scores and multiple engagement or temporal features in the dataset. The Student Scores to Overall Clicks reveal a positive association between total VLE clicks and student scores, suggesting that higher engagement levels are generally linked to better academic performance. The Student Scores to Assessment Weight indicates that most assessments have lower weightings, with a dense concentration of points near zero weight and minimal variation in scores across higher weights. The Student Scores to Day Submitted shows a weak negative trend, suggesting that later submission dates may be modestly associated with lower scores, though the relationship appears minimal. The Impact of Submission

Lateness on Performance illustrates a clearer negative relationship, where increased lateness correlates with lower scores, as captured by the downward-sloping regression line. Collectively, these plots highlight that engagement intensity is a strong positive predictor, while lateness is a notable negative factor influencing student performance, whereas assessment weight and submission timing exhibit weaker or more diffuse relationships.

C. Data Preprocessing

The data preprocessing is a critical stage in predictive modeling, ensuring that raw educational data is transformed into a structured, consistent, and analytically suitable format. This process involves cleaning and standardizing heterogeneous data sources, handling missing or inconsistent values, and encoding variables to align with the input requirements of advanced deep learning architectures.

Data cleaning is performed first to remove noise, inconsistencies, and missing values from the dataset. For the raw dataset $= \{x_i, y_i\}^N_{i=1}$, where x_i is the feature vector and y_i, the corresponding label for the i^{th} student, the missing values in a numerical feature f_j are replaced using mean imputation, as in Eq. (1):

$$f_j^* = \frac{\sum_{i=1}^{N} f_{ij}}{N_j^{obs}} \quad (1)$$

where, N_j^{obs} represents the number of observations without missing values for feature f_j, and f_j^* is the imputed value. For categorical variables, the missing entries were replaced by the mode, as in Eq. (2):

$$f_j^* = arg \underset{c \in C_j}{max\ count\ (c)} \quad (2)$$

where, C_j is the set of possible categories for f_j. Outliers in continuous variables are detected using the z-score method, as in Eq. (3):

$$z_{ij} = \frac{f_{ij} - \mu_j}{\sigma_j} \quad (3)$$

where, μ_j and σ_j denote the mean and standard deviation of feature f_j, respectively. Instances with $|z_{ij}| > 3$ are flagged for removal or transformation. Following data cleaning, feature engineering is performed to transform the dataset into a structured form compatible with the model. Categorical variables such as gender, region, and highest education were encoded using one-hot encoding, producing binary indicator vectors for each observation. For a categorical feature f_j with K_j distinct categories, each observation i is represented, as in Eq. (4):

$$f_{i,j}^{(k)} = \begin{cases} 1\ ;if\ observation\ i\ belongs\ to\ category\ k \\ 0\ ;otherwise \end{cases} \forall k \in [1, K_j] \quad (4)$$

Numerical attributes, including click counts, assessment scores, and time-on-task, were normalized to a uniform scale to ensure balanced gradient updates during model training. The min–max transformation is applied, as in Eq. (5):

$$f'_{i,j} = \frac{f_{i,j} - min\ (f_j)}{max(f_j) - min\ (f_j)} \quad (5)$$

where, $f'_{i,j}$ is the normalized value for feature f_j in observation i. To capture engagement dynamics, temporal VLE logs were aggregated into weekly indicators of student activity. The weekly engagement score for student i in week w was computed, as in Eq. (6):

$$E_{i,w} = \frac{\sum_{t \in w} clicks_{i,t}}{\Delta t_w} \quad (6)$$

where, $clicks_{i,t}$ is the total VLE interactions at timestamp t and Δt_w is the number of active days in week w. In addition, assessment-related features were derived by computing the weighted average of scores, as in Eq. (7):

$$A_{(i)} = \frac{\sum_{k=1}^{K} w_k \cdot m_{(i,k)}}{\sum_{k=1}^{K} w_k} \quad (7)$$

where, $m_{(i,k)}$ denotes the mark of student i in assessment k and w_k is its respective weight. Temporal score change was measured, as in Eq. (8), capturing improvement or decline between consecutive assessments.

$$\Delta A_{(i,t)} = m_{(i,t)} - m_{(i,t-1)} \quad (8)$$

The final engineered feature matrix X^* integrates demographic encodings, normalized continuous attributes, weekly engagement profiles, and assessment-based indicators, providing a comprehensive, multi-view representation of each learner. For the score range prediction task, continuous marks were discretized into performance bands to align with the feedback generation module. The score band for student i is as in Eq. (9):

$$Band_i = \begin{cases} 0\ ;if\ 0 \le S_i < 40 \\ 1\ ;if\ 40 \le S_i < 60 \\ 2\ ;if\ 60 \le S_i < 80 \\ 3\ ;if\ 80 \le S_i \le 100 \end{cases} \quad (9)$$

where, S_i denotes the final computed score from all available assessments. These engineered features form the input vector X_i for subsequent representation learning in the proposed framework. Feature selection is further performed to retain critical informative predictors and reduce model complexity. The mutual information criterion is applied between each feature f_j and the target variable y, as in Eq. (10):

$$MI(f_j, y) = \sum_{f_j} \sum_{y} p(f_j, y)\ log \frac{p(f_j, y)}{p(f_j)\ p(y)} \quad (10)$$

where, $p(f_j, y)$ is the joint probability distribution of feature f_j and target y and $p(f_j), p(y)$ are the respective marginal probabilities. Features with $MI(f_j, y) < \tau$ are discarded, where τ is the data-driven threshold. The processed dataset X^* is then split into training and testing subsets in the ratio 80:20. Class imbalance handling is applied only to the training set to prevent information leakage. Using the Synthetic Minority Oversampling Technique (SMOTE), new synthetic samples are generated, as in Eq. (11):

$$x_{new} = x_i + \lambda \cdot (x_{nn} - x_i) \quad (11)$$

where, x_i is a minority class sample, x_{nn} is one of its KNN and $\lambda \in [0,1]$ is a random interpolation factor. Finally, the balanced training set is structured for the proposed architecture, with each observation X_i represented, as in Eq. (12):

$$X_i = [X_i^{demo}, X_i^{eng}, X_i^{asses}, X_i^{temp}] \quad (12)$$

where, X_i^{demo} contains demographic encodings, X_i^{eng} with weekly engagement scores, assessment-derived metrics in X_i^{asses} and X_i^{temp} contains temporal change indicators to facilitate multi-view input compatibility with the hierarchical attention encoder.

D. Model Deployment

1) TabNet architecture: TabNet is a DL architecture designed specifically for tabular data, combining the interpretability of DTs with the representational power of NNs [24]. The model processes data in multiple sequential decision steps, where at each step a subset of the most relevant features is selected through an attentive feature-masking mechanism. The process begins with an input vector $X_i \in \mathbb{R}^d$ for student i, which is first normalized and passed through a shared feature transformer network $FT_{shared}(\cdot)$. The initial hidden representation is as in Eq. (13):

$$H_i^{(0)} = FT_{shared}(X_i; W_f) \quad (13)$$

where, W_f are learnable weights of the shared feature transformer block. At each decision step $t \in \{1,2,\dots,T\}$, TabNet computes an attention mask $M^{(t)}$ over the features using an Attentive Transformer. The mask is modulated by a prior scale vector $P^{(t)} \in [0,1]^d$ which controls feature reuse and is initialized as $P^{(t)} = 1$ for $t = 1$. The mask is given as in Eq. (14):

$$M_i^{(t)} = sparsemax(P^{(t)} \odot AT^{(t)}(H_i^{(t-1)}; W_a)) \quad (14)$$

where, W_a is the attention weight matrix, $\odot$ denotes element-wise multiplication, and the sparsemax activation ensures that only a subset of features receive non-zero weights, enhancing interpretability. The masked feature vector is obtained as in Eq. (15) and passed through the decision step network as in Eq. (16):

$$\tilde{H}_i^{(t)} = M_i^{(t)} \odot H_i^{(t-1)} \quad (15)$$

$$(D_i^{(t)}, H_i^{(t)}) = DecisionBlock^{(t)}(\tilde{H}_i^{(t)}; W_d) \quad (16)$$

where, $D_i^{(t)} \in \mathbb{R}^k$ is the decision output contributing directly to the prediction and $H_i^{(t)}$ is the transformed feature representation for the current step. The prior scale vector for the next step is updated to reduce the weight of features already selected, allowing partial reuse via the relaxation parameter $\gamma > 1$ as in Eq. (17):

$$P^{(t+1)} = P^{(t)} \odot (\gamma - M_i^{(t)}) \quad (17)$$

The final prediction is the sum of decision outputs across all T steps as in Eq. (18):

$$\hat{y}_i = \sum_{t=1}^{T} D_i^{(t)} \quad (18)$$

In the proposed framework, however, instead of using only $\hat{y}_i$ from the aggregated decision outputs, a latent representation is formed by concatenating the transformed hidden vectors from all decision steps as in Eq. (19):

$$Z_i = Concat(H^{(1)}, H^{(2)}, \dots, H^{(T)}) \quad (19)$$

The enriched feature vector Z_i is then fed into the XGBoost model that refines the prediction of the student's score range and enables sharper decision boundaries. The basic architecture of TabNet is depicted in Fig. 6.

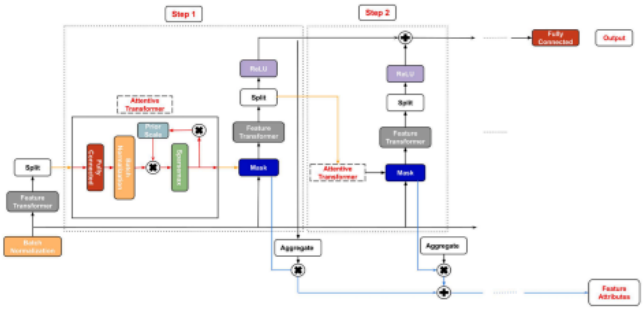

Fig. 6. Basic architecture of TabNet.

2) XGBoost Regressor: Extreme Gradient Boosting (XGBoost) is an optimized implementation of the gradient boosting framework designed for scalability, efficiency, and high predictive accuracy on structured data [23]. The algorithm builds an ensemble of decision trees sequentially, where each new tree f_t is trained to minimize the residual errors of the previous trees. For the dataset $\{(X_i, y_i)\}_{i=1}^{n}$, the model prediction at iteration t is as in Eq. (20):

$$\hat{y}_i = \hat{y}_i^{(t-1)} + \eta f_t(Z_i) \quad (20)$$

where, η is the learning rate and f_t represents the freshly added regression tree. The optimization process minimizes regularized objective function, as shown in Eq. (21):

$$L(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t) \quad (21)$$

where, $l(y_i, \hat{y}_i)$ is the differentiable loss function and $\Omega(f_t)$ is the regularization term defined, as shown in Eq. (22):

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2 \quad (22)$$

where, T is the number of leaves in tree f_t, w_j is the weight assigned to leaf j, γ penalizes the creation of excessive leaves to control the model complexity, and λ controls L2 regularization on leaf weights.

Fig. 7 represents the basic architecture of XGBoost improves traditional gradient boosting through features such as parallelized tree construction, sparsity-aware split finding and weighted quantile sketch for efficient handling of missing values. Its ability to model complex, non-linear relationships makes it highly effective for educational datasets where interactions between demographic, engagement and assessment features can strongly influence performance predictions.

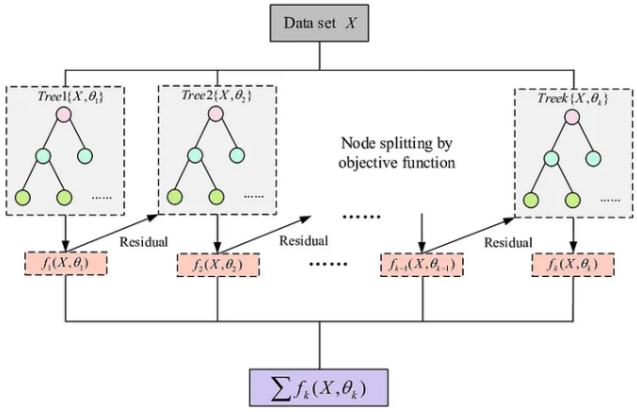

Fig. 7. Basic architecture of XGBoost Regressor.

3) Proposed TabNet-XGBoost hybrid model: In the proposed TabNet-XGBoost architecture, TabNet serves as the representation learning module, extracting a compact yet information-rich latent feature vector from the pre-processed student data. The output of the TabNet module, the concatenated vector Z_i, captures both global and stepwise feature importance patterns, effectively encoding demographic, engagement and assessment-related information into a single high-dimensional embedding. This enriched feature vector Z_i is then passed into the XGBoost Regressor, which acts as the final prediction layer. In regression mode, XGBoost iteratively builds trees to refine the score prediction. After convergence, the continuous score prediction is discretized into four score bands representing performance ranges. This integration allows TabNet to perform interpretable and sparsity-driven feature selection, while XGBoost captures non-linear relationships and sharpens class separation boundaries. The combination results in a robust hybrid model capable of predicting students' score ranges with high precision while retaining interpretability for SHAP-based feedback generation in later stages.

Following score band prediction, the model incorporates SHapley Additive exPlanations (SHAP) to ensure interpretability and to support the customized feedback generation module. SHAP, grounded in cooperative game theory, assigns each feature a Shapley value ϕ_j representing its average marginal contribution to the model's prediction across all possible feature subsets. For a given student i with a feature set F and model prediction $f(X_i)$ the Shapley value for feature j s computed as in Eq. (23):

$$\phi_j = \sum_{S \subseteq F\{j\}} \frac{|S|!\,(|F|-|S|-1)!}{|F|!} \left[f_{S\cup\{j\}}(X_i) - f_s(X_i)\right] \quad (23)$$

where, S is a subset of features excluding j, $f_{S\cup\{j\}}(X_i)$ is the model output when j is included and $f_s(X_i)$ is the output without j. This ensures a fair and coherent measure of feature importance, regardless of ordering or correlation. In the proposed hybrid framework, SHAP is applied after the XGBoost stage, taking the enriched latent vector Z_i as the feature input and the XGBoost model's predicted score range as the target for explanation [24]. Following SHAP analysis, the feature importance values for each individual student are mapped onto seven pre-defined feedback domains: Engagement, Assessment Preparedness, Time Management, Learning Resource Utilization, Interaction Frequency, Content Comprehension and Consistency of Effort. For each domain d, the mean SHAP value $s_{i,d}$ is computed by averaging SHAP contributions of features belonging to that domain as in Eq. (24):

$$s_{i,d} = \frac{1}{|F_d|}\sum_{f \in F_d} SHAP_{i,f} \quad (24)$$

where, F_d is the set of features assigned to the domain d. Domains with high positive SHAP values indicate areas that most contributed to predicted high performance, while high negative SHAP values signal weaknesses requiring intervention. A feedback table is then generated for each student, with rows representing the domains and columns indicating performance status along with the recommendations. This integration ensures that the system not only predicts student performance with high accuracy but also delivers interpretable, actionable and structured recommendations to improve learning outcomes. The algorithm for the proposed model is as shown below (see Algorithm 1).

Algorithm 1: TabNet-XGBoost hybrid model for Student performance prediction and customized feedback

Input:

- *OULAD dataset* $D = \{(X_i, y_i)\}_{i=1}^{N}$
- *Y: Label vector corresponding to X*
- *Label vector* $\in \{0,1,2,3\}$*, each value for different score range*

Output:

- *Predicted class label* $Y \in \{0,1,2,3\}$ *and customized feedback*

Begin:

Data collection

- *Load OULAD dataset*
- *Extract feature matrix X and label vector Y*

Data Preprocessing and Feature extraction

- *Missing values imputation:*

$$f_j^* = \frac{\sum_{i=1}^{N} f_{ij}}{N_j^{obs}}$$

- *Outliner detection and handling:*

$$z_{ij} = \frac{f_{ij} - \mu_j}{\sigma_j}, remove\ if\ |z_{ij}| > 3$$

- *One hot encoding for categorical features:*

$$f_{i,j}^{(k)} = \begin{cases} 1\,; if\ observation\ i\ belongs\ to\ category\ k \\ 0\,; otherwise \end{cases}$$

- *Normalization:*

$$f'_{i,j} = \frac{f_{i,j} - \min(f_j)}{\max(f_j) - \min(f_j)}$$

- *Various score computations*

Weekly engagement score, $E_{i,w} = \frac{\sum_{t \in w} clicks_{i,t}}{\Delta t_w}$

Weighted Assessment Score, $A_{(i)} = \frac{\sum_{k=1}^{K} w_k \cdot m_{(i,k)}}{\sum_{k=1}^{K} w_k}$

Temporal score change, $\Delta A_{(i,t)} = m_{(i,t)} - m_{(i,t-1)}$

- *Discretize score into performance bands*

$$Band_i = \begin{cases} 0\,; if\ 0 \le S_i < 40 \\ 1\,; if\ 40 \le S_i < 60 \\ 2\,; if\ 60 \le S_i < 80 \\ 3\,; if\ 80 \le S_i \le 100 \end{cases}$$

- *Train-Test Split: Split into 80:20 ratio* $X_{train}, X_{test}, Y_{train}, Y_{test} = train\_test(X, Y, test_{size} = 0.2)$

TabNet

- *Transform input:* $H_i^{(0)} = FT_{shared}(X_i; W_f)$
- *For each decision step t:*
  a. *Compute attention mask,* $M_i^{(t)} = sparsemax\,(P^{(t)} \odot AT^{(t)}(H_i^{(t-1)}; W_a)$
  b. *Apply mask and process through decision block,* $(D_i^{(t)}, H_i^{(t)}) = DecisionBlock^{(t)}(\widetilde{H}_i^{(t)}; W_d)$
  c. *Update the prior scale,* $P^{(t+1)} = P^{(t)} \odot (\gamma - M_i^{(t)})$
- *Concatenate hidden states:* $Z_i = Concat(H^{(1)}, H^{(2)}, \ldots, H^{(T)})$

XGBoost prediction

- *Initialize Predictions,* $\hat{y}_i^{(0)} = 0$
- *For* $t = 1\ to\ T$
  a. *Fit tree* f_t *to residuals*
  b. *Update prediction:* $\hat{y}_i = \hat{y}_i^{(t-1)} + \eta f_t(Z_i)$
- *Optimize regularized objective*

$$L(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t)$$

SHAP based interpretation

- *Apply SHAP to* Z_i *and* $\hat{y}_i$ *to compute per dpomain feature importance values*

$$\phi_j = \sum_{S \subseteq F\{j\}} \frac{|S|!\ (|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{j\}}(X_i) - f_s(X_i) \right]$$

Feedback Generation

- *For each domain d, map the SHAP score to feedback category*

$$s_{i,d} = \frac{1}{|F_d|} \sum_{f \in F_d} SHAP_{i,f}$$

Model Compilation and Training

- *Compile model with loss = sparse categorical crossentropy, learning rate = 0.001, optimizer = Adam, Epochs =50*
- *Train model: model.fit (X_train, y_train)*

Evaluation and Model Saving

- *Evaluate model: model. evaluate (X_test, Y_test)*
- *Tune hyperparameters*
- *Save the model*

End

E. Simulation Setup

The proposed TabNet-XGBoost hybrid model was implemented using a high-performance computational environment. The system configuration included an Intel Core i7 processor, an NVIDIA GeForce GTX 1080Ti GPU and 32 GB of RAM that collectively ensured efficient handling of the intensive training and evaluation processes involved in IoT security anomaly identification and classification. To develop the model, Keras API built on TensorFlow and python was selected as the programming language. Google Colaboratory (Colab) was used for model training and testing, taking advantage of its free access to powerful GPUs and cloud-based execution environment that improved the study's accessibility and reproducibility. The hyperparameters that affect the model behavior significantly are manually selected before training and have a direct impact on the framework's rate of convergence, generalization capability and ultimate classification performance. The full list of hyperparameters and training settings employed in the study is summarized in Table I.

TABLE I. HYPERPARAMETER SPECIFICATIONS

Hyperparameters	Values
Epochs	50
Dropout	0.2
Activation function	ReLU
Optimizer	ADAM
Loss function	Sparse categorical cross entropy
Batch size	32
Learning Rate	0.001

IV. RESULTS AND DISCUSSION

A set of standard evaluation metrics has been employed to evaluate the performance of the proposed model, as illustrated in Eq. (25) to Eq. (28). These measures are mathematically computed using the core elements of the confusion matrix: True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN). Accuracy indicates the overall correctness while recall and precision highlight the framework's effectiveness in the prediction of score ranges without many misses or false alarms.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (25)$$

$$Precision = \frac{TP}{TP+FP} \quad (26)$$

$$Recall = \frac{TP}{TP+FN} \quad (27)$$

$$F1 - score = 2 \times \frac{precision \times Recall}{Precision + Recall} \quad (28)$$

Accuracy visualizes a model's learning progress across training epochs, showing how well the model is fitting the data. The accuracy plot indicates improvements in predictive performance over time, while the loss plot reflects the model's error reduction. Comparing training and validation curves helps identify overfitting, underfitting or stable convergence. These insights guide hyperparameter tuning, regularization adjustments and architecture refinements to improve model generalization and performance.

The accuracy plot in Fig. 8 demonstrates a consistent improvement in both training and validation accuracy over the 50 epochs, starting from around 0.90 and 0.87, respectively. Both curves rise steadily, with validation accuracy closely tracking training accuracy, peaking at approximately 0.989 for training and 0.988 for validation.

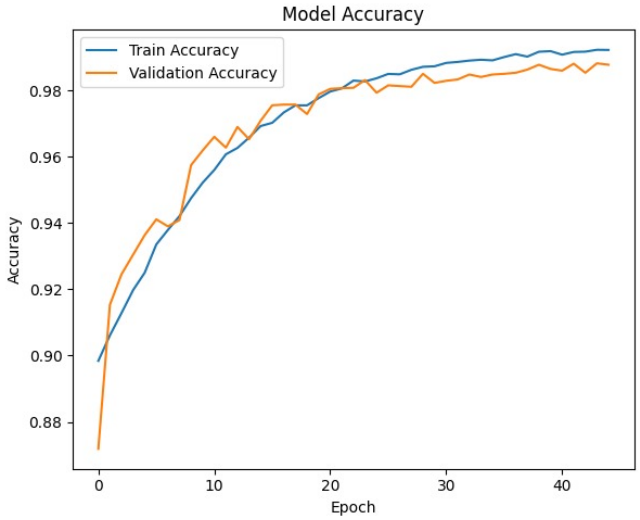

Fig. 8. Accuracy plot.

The minimal gap between the curves suggests that the hybrid architecture generalizes well to unseen data and refrains from significant overfitting. This strong and parallel upward trend indicates that the chosen architecture and hyperparameters are effective in progressively improving classification performance throughout training.

Fig. 9 illustrates the confusion matrix that indicates the proposed model's remarkable classification performance across all four score bands. Misclassifications are minimal and primarily occur between adjacent score bands, such as 0–Less than 40 and 40–Less than 60, suggesting occasional boundary overlap in predicted scores. Overall, the distribution reflects both high accuracy and stability in multi-class categorization.

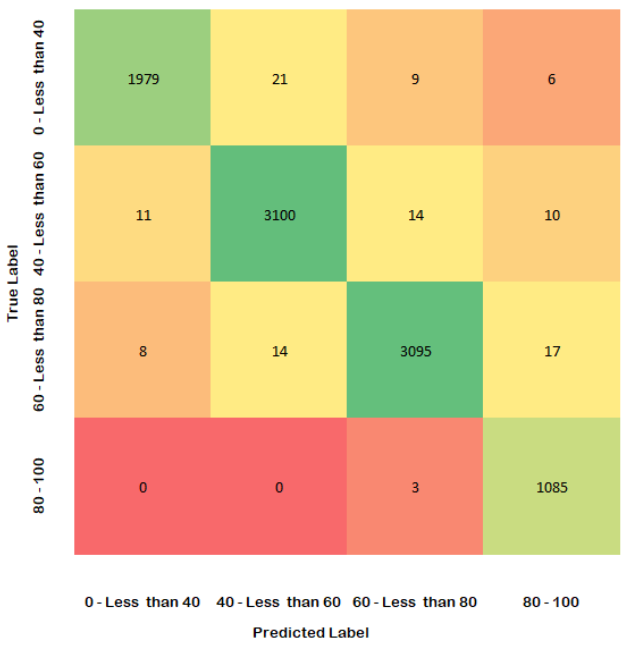

Fig. 9. Confusion matrix.

The metric evaluation results in Fig. 10 demonstrate the robustness and effectiveness of the TabNet-XGBoost hybrid model across all performance indicators.

An accuracy of 98.79%, indicated that the vast majority of predictions correctly matched the actual class labels across the four score bands. The precision value of 98.54% reflects the model's ability to minimize FPs, guaranteeing that most of the predicted instances for each class were indeed correct. The recall score of 98.89% shows the correct identification of true instances across all classes, with very few false negatives.

The F1-score of 98.71%, harmonically balances precision and recall, confirms the framework's consistent performance in both detecting true cases and avoiding misclassifications. Collectively, these findings highlight that the TabNet-XGBoost architecture maintains a balanced trade-off between precision and recall while achieving excellent overall classification performance.

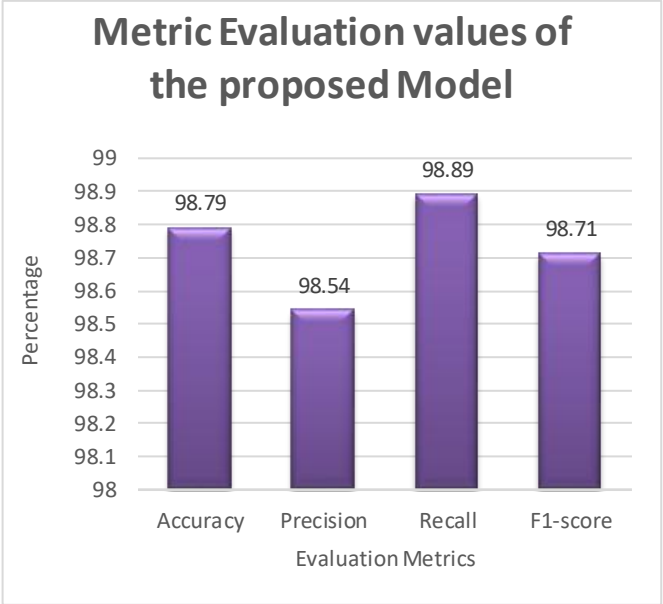

Fig. 10. Metric evaluation values.

Deep learning architectures such as LSTM, RNN and Conv-LSTM improve temporal dependency modeling, achieving accuracy levels exceeding 90%, but these models tend to suffer from high computational costs, longer training times and limited interpretability. Transformer-based and attention-augmented models address feature importance explicitly, yet still struggle with optimizing for structured tabular data, where sparsity and heterogeneous feature types prevail. In contrast, the proposed TabNet–XGBoost hybrid model leverages TabNet's interpretable feature selection capabilities with XGBoost's powerful non-linear decision boundaries, achieving an accuracy of 98.8%. This not only surpasses the performance of prior models but also offers enhanced interpretability and adaptability to varied educational datasets, making it a robust choice for real-world deployment. Fig. 11 represents graphical representation of the accuracy comparison.

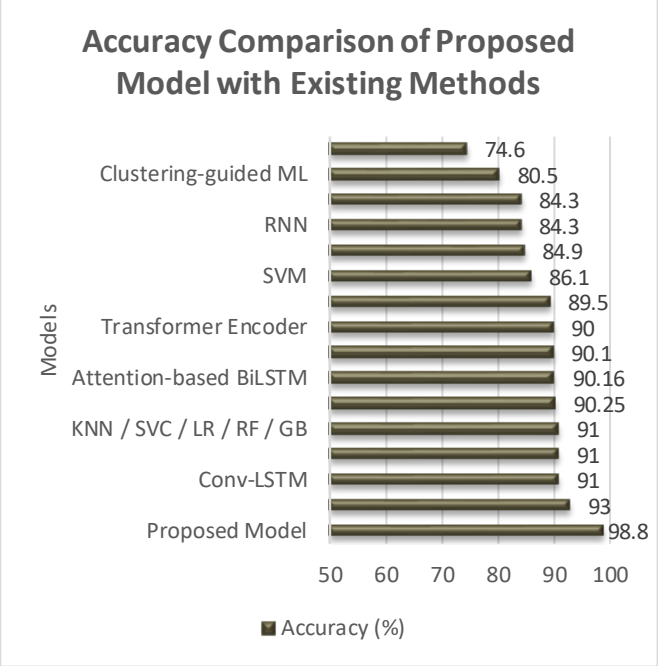

Fig. 11. Accuracy comparison.

V. Conclusion

This study presented a hybrid TabNet–XGBoost framework for predicting student performance and providing targeted feedback, demonstrating the potential of combining deep learning-based feature selection with powerful gradient boosting techniques for educational data mining. By integrating TabNet's attentive feature-masking mechanism, the model efficiently identified and utilized the most informative attributes from the OULAD, while XGBoost refined the latent feature representations to enhance score range classification accuracy The incorporation of SHAP enabled model interpretability by quantifying the contribution of each feature to individual predictions, facilitating the development of a feedback module tailored to seven pedagogical domains. The proposed architecture achieved an accuracy of 98.8%, significantly outperforming conventional ML and DL baselines. Furthermore, the interpretability offered by SHAP ensured that predictions were not only highly accurate but also actionable, aligning predictive analytics and instructional decision-making. Future works may focus on expanding the feature space to incorporate additional behavioral, social and temporal engagement indicators, as well as validating the framework across multiple institutions.

References

[1] Caprara, L., & Caprara, C. (2022). Effects of virtual learning environments: A scoping review of literature. *Education and information technologies*, *27*(3), 3683-3722.

[2] Turan, Z., Kucuk, S., & Cilligol Karabey, S. (2022). The university students' self-regulated effort, flexibility and satisfaction in distance education. *International Journal of Educational Technology in Higher Education*, *19*(1), 35.

[3] Caraig, R. V., & Fabro, R. G. (2022). Students' engagement and study habits in the online distance learning environment. *Int. J. Sci. Res. in Multidisciplinary Studies Vol*, *8*(9).

[4] Carby, N. (2023). Personalized feedback in a virtual learning environment. *Journal of Educational Supervision*, *6*(1), 36.

[5] Almalki, A. D. A., & Mohammed, A. I. (2022). The Effect of Immediate and Delayed Feedback in Virtual Classes on Mathematics Students' Higher Order Thinking Skills. *Journal of Positive School Psychology*, *6*(6).

[6] Ismanto, E., Ghani, H. A., & Saleh, N. I. B. M. (2023). A comparative study of machine learning algorithms for virtual learning environment performance prediction. *IAES International Journal of Artificial Intelligence*.

[7] Torkhani[1], W., & Rezgui, K. (2025, February). OULAD MOOC Student Performance Prediction using Machine and Deep Learning. In *Proceedings of International Conference on Decision Aid and Artificial Intelligence (ICODAI 2024)* (Vol. 12, p. 228). Springer Nature.

[8] Liu, Y., Fan, S., Xu, S., Sajjanhar, A., Yeom, S., & Wei, Y. (2022). Predicting student performance using clickstream data and machine learning. *Education Sciences*, *13*(1), 17.

[9] Arashpour, M., Golafshani, E. M., Parthiban, R., Lamborn, J., Kashani, A., Li, H., & Farzanehfar, P. (2023). Predicting individual learning performance using machine-learning hybridized with the teaching-learning-based optimization. *Computer Applications in Engineering Education*, *31*(1), 83-99.

[10] Rao, G. M., & Kumar, P. K. K. (2021). Students performance prediction in online courses using machine learning algorithms. *United Int. J. Res. Technol*, *2*(11), 74-79.

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

[12] Yu, Chih-Chang & Wu, Leon. (2021). Early Warning System for Online STEM Learning—A Slimmer Approach Using Recurrent Neural Networks. Sustainability. 13. 12461. 10.3390/su132212461.

[13] Kusumawardani, S. S., & Alfarozi, S. A. I. (2023). Transformer encoder model for sequential prediction of student performance based on their log activities. *Ieee Access*, *11*, 18960-18971.

[14] Chen, H. C., Prasetyo, E., Tseng, S. S., Putra, K. T., Kusumawardani, S. S., & Weng, C. E. (2022). Week-wise student performance early prediction in virtual learning environment using a deep explainable artificial intelligence. *Applied Sciences*, *12*(4), 1885.

[15] Adnan, M., Habib, A., Ashraf, J., Mussadiq, S., Raza, A. A., Abid, M., ... & Khan, S. U. (2021). Predicting at-risk students at different percentages of course length for early intervention using machine learning models. *Ieee Access*, *9*, 7519-7539.

[16] Riestra-González, M., del Puerto Paule-Ruíz, M., & Ortin, F. (2021). Massive LMS log data analysis for the early prediction of course-agnostic student performance. *Computers & Education*, *163*, 104108.

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

[18] Chen, Y., Wei, G., Liu, J., Chen, Y., Zheng, Q., Tian, F., ... & Wu, Y. (2023). A prediction model of student performance based on self-attention mechanism. *Knowledge and Information Systems*, *65*(2), 733-758.

[19] Hakkal, S., & Ait Lahcen, A. (2024). XGBoost to enhance learner performance prediction. *Computers and Education: Artificial Intelligence*, *7*, 100254.

[20] Iatrellis, O., Savvas, I. K., Fitsilis, P., & Gerogiannis, V. C. (2021). A two-phase machine learning approach for predicting student outcomes. *Education and Information Technologies*, *26*(1), 69-88.

[21] https://www.kaggle.com/datasets/anlgrbz/student-demographics-online-education-dataoulad

[22] Arik, S. Ö., & Pfister, T. (2021, May). Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 35, No. 8, pp. 6679-6687).

[23] Le, T. T. H., Oktian, Y. E., & Kim, H. (2022). XGBoost for imbalanced multiclass classification-based industrial internet of things intrusion detection systems. *Sustainability*, *14*(14), 8707.

[24] Ergün, S. (2023). Explaining XGBoost predictions with SHAP value: A comprehensive guide to interpreting decision tree-based models. *New Trends in Computer Sciences*, *1*(1), 19-31.