Journal of Computer and Electrical Sciences (JCES)- Volume-1 Issue-4, April 2026
ISSN: 3049-2602

A HYBRID PREDICTIVE MODELING APPROACH FOR EARLY STUDENT PERFORMANCE ESTIMATION USING SEQUENTIAL LEARNING ACTIVITY DATA

K. Lakshmi Srihitha [1], V. Sri Lakshmi [2], V. Kavya Sri [3], G. Bhavya Sai [4],

Dr. K. Siva Rama Krishna [5]

[1,2,3,4]Student, Department of Computer Science And Engineering(Data Science)

Andhra Loyola Institute Of Engineering And Technology,Vijayawada, Andhra Pradesh, India
[5]*Associate Professor, Department of Computer Science And Engineering(Data Science)*
Andhra Loyola Institute Of Engineering And Technology
Andhra Loyola Institute Of Engineering And Technology, *vijayawada,*
Andhra Pradesh, India
Email id:klakshmisrihitha@gmail.com, lakshmivinukonda99@gmail.com,
vkavyasri20@gmail.com,gottumukkalabhavya@gmail.com,
sivaramakosuru@gmail.com

Abstract: *Early identification of students at academic risk is one of the most actionable goals in learning analytics, as interventions are more effective when implemented before outcomes are fixed. This study presents a hybrid prediction pipeline built on the Open University Learning Analytics Dataset (OULAD), combining two complementary perspectives of student behavior: engineered tabular indicators and week-level activity sequences. The tabular branch uses a Random Forest model with assessment and VLE summary features, while the temporal branch uses an LSTM model over weekly click activity. Final predictions are produced through weighted probability fusion, and then converted into risk bands for intervention planning. The full pipeline covers data preparation, feature construction, model training, evaluation, visualization, and reporting. This design aims to balance practical interpretability with the ability to capture behavioral change over time. The contribution is methodological and operational: we describe an end-to end workflow that institutions can adapt, together with explicit design choices for class imbalance, temporal modeling, and risk-oriented thresholding rather than accuracy-only optimization.*

Keywords: Educational Data Mining, Learning Analytics, Student Success Prediction, Random Forest, LSTM, Hybrid Modeling, Early Warning Systems.

1. INTRODUCTION

Online learning platforms continuously record how students interact with course material, assessments, and digital resources [6], [9]. These interaction traces are valuable for predicting course outcomes before the end of the term, which can help institutions provide timely academic support [4], [7]. However, many prediction systems rely on a single representation of behavior: either static aggregate features or fully sequential models.

This creates a familiar trade-off. Aggregate models are often easier to explain to instructors, but they can miss shifts in engagement patterns during the term. Sequence models capture temporal dynamics more naturally, yet they are usually harder to interpret in day-to-day educational settings [4], [8]. To address this gap, the present work combines both views: a Random Forest branch for interpretable aggregate patterns [2] and an

LSTM branch for week-by-week behavioral trajectories [3]. Prior EDM research supports the value of combining multiple representations for stronger and more stable prediction [5], [10].

The objective is not limited to accuracy alone. The system is designed to produce operational risk categories (low, medium, high) that academic teams can use to triage interventions. In that sense, the paper contributes a reproducible end-to-end workflow that links model development with deployment-oriented outputs for early-warning practice [4], [8], [18].

Contributions. This paper makes the following contributions: (1) a dual-branch architecture that fuses tabular and sequential signals with a transparent weighting scheme suitable for tuning by stakeholders; (2) a concrete feature engineering recipe aligned with OULAD's multi-table structure, including assessment summaries and fixed-horizon weekly click tensors for recurrent modeling; (3) an intervention-oriented risk mapping from fused probabilities to discrete bands, emphasizing recall-oriented use cases common in advising; and (4) a reporting-oriented pipeline that pairs essential exploratory analysis with model diagnostics appropriate for learning analytics reporting.

The remainder of the paper is organized as follows. Section 2 situates the approach within prior work on student performance prediction and hybrid modeling. Section 3 describes the dataset and the binary formulation of the prediction task. Section 4 details methodology, including preprocessing, models, fusion, evaluation, and the experimental protocol. Sections 5 and 6 present exploratory findings and experimental outputs. Section 7 discusses limitations and deployment implications, and Section 8 concludes.

2. Related Work

Student success prediction from LMS data. A long line of research uses log data from learning management systems to estimate grades, dropout risk, or final certification outcomes [6], [7], [9]. Classical approaches often rely on counts of actions, time-on-task, forum participation, and assessment statistics because these features are straightforward to compute and align with instructors' intuition. Surveys in educational data mining and learning analytics emphasize that predictive models should be evaluated not only by statistical metrics but also by their fit to institutional processes such as advising workflows and resource constraints [4], [8].Temporal and sequential modeling. When engagement changes across weeks, recurrent models and related sequence architectures provide a natural inductive bias for ordered clickstreams or session-level histories [3]. In practice, sequence models can track "momentum" in activity that aggregate totals obscure—for example, a student whose clicks decline sharply after midterm may differ from one with steady participation despite similar cumulative counts. At the same time, deep sequence models can overfit smaller cohorts and may require careful regularization and early stopping [14], [15].

Ensembles and hybrid models. Ensemble methods such as Random Forests remain strong baselines for tabular educational data because they capture non-linear interactions and provide feature importance estimates that support explanation [2]. Hybrid systems that combine heterogeneous feature sets or model families have been explored to improve robustness: predictions may stabilize when one representation is noisy while another remains informative [5], [10]. Our work follows this intuition by maintaining separate branches and combining their calibrated outputs rather than forcing all signals into a single representation prematurely.Gap addressed. Many published pipelines emphasize either interpretable shallow models or end-to-end deep learning, but give less attention to how outputs should be converted into operational risk strata and reviewed alongside confusion structures that matter for early warning [6], [16]. We therefore foreground late fusion, risk-band mapping, and evaluation framing aligned with intervention cost asymmetries.

3. Dataset and Problem Formulation

The experiments use OULAD, a multi-table benchmark dataset that includes student profile attributes, registration history, assessment records, and VLE interaction logs [1]. OULAD is widely

used because it offers realistic scale, multiple presentations of modules, and rich behavioral traces while remaining publicly available for reproducible comparison. The dataset spans numerous modules and student–module registrations, which supports analyses that are not dominated by a single course idiosyncrasy; at the same time, heterogeneity across modules implies that models may need regularization and careful evaluation so that performance reflects generalizable signals rather than memorized module-specific quirks. Data integration is performed using the following source tables:

- `studentInfo.csv`
- `studentRegistration.csv`
- `studentAssessment.csv`
- `assessments.csv`
- `studentVle.csv`
- `vle.csv`

Unit of analysis and keys. Modeling is carried out at the student–module presentation level when appropriate, consistent with how OULAD records registrations and outcomes: each row ties a student to a specific module instance, allowing both static attributes and term-specific behavior to be aligned. Joins preserve identifiers across tables so that assessment events and VLE events aggregate to the same analytical unit used for labeling.

The target variable is final_result, reformulated as a binary prediction task:

- `Pass` and `Distinction` -> 1 (success)
- `Fail` and `Withdrawn` -> 0 (non-success)

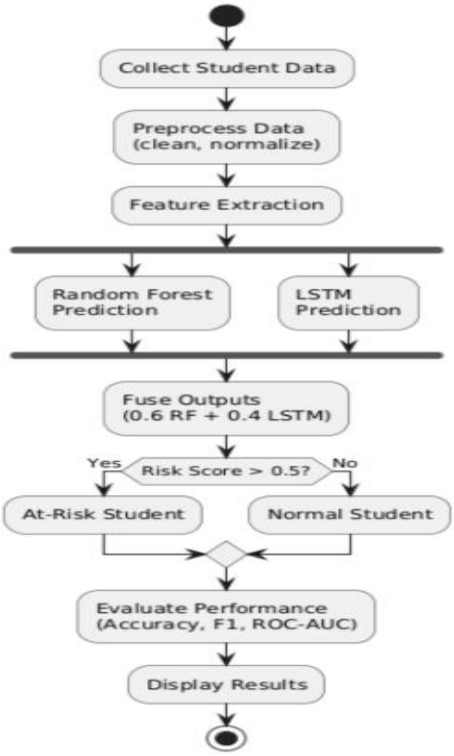
This grouping reflects an early-warning perspective, where both failing and withdrawing outcomes represent cases requiring proactive support [4], [6]. Binary reformulation also simplifies thresholding and reporting for non-technical stakeholders, although it collapses nuanced performance levels; we return to this limitation in the discussion.

Preprocessing principles. Before feature construction, records are filtered to coherent registrations with usable activity traces where required by the modeling branch. Identifiers are validated to prevent leakage across split boundaries when students appear in multiple presentations. Categorical fields are encoded in a manner consistent with the tree-based model, while numeric fields are scaled where needed for stable training in the deep branch.

4. Methodology

4.1System Architecture

The framework follows a dual-branch architecture with late fusion.



The workflow is organized into five stages:

1. Load and validate the required OULAD tables.
2. Construct aggregate and sequential student representations.
3. Train Random Forest and LSTM models independently.
4. Fuse branch probabilities with `p_hybrid = 0.6 p_rf + 0.4 p_lstm`.
5. Convert fused probabilities into risk bands and export decision outputs.

Late fusion keeps branch training modular: each model can be maintained, audited, and replaced with less coupling than a single joint network that entangles representations early. The weights (0.6 / 0.4) reflect a practical prior that interpretable aggregate indicators remain strong on tabular educational data, while sequences provide a corrective signal when weekly trajectories diverge from what aggregates suggest.

Institutions may tune these weights or learn them via validation when sufficient labeled data exist.

4.2 Data Integration and Feature Engineering

Student-level samples are formed by joining profile and registration data with assessment and VLE logs using consistent keys. These logs are transformed into model-ready features:

- Assessment features: `assessment_count`,
- `avg_score`, `avg_delay`
- VLE aggregate features: `total_clicks`, `active_days`,
- `avg_clicks`
- VLE sequential features: weekly click bins (`max_weeks = 30`)

Assessment features summarize timeliness and performance on graded work delays capture procrastination and pacing, which are frequently associated with risk in distance learning contexts [6], [9]. Aggregate VLE features summarize the intensity and regularity of platform use. For sequence modeling, weekly click counts are reshaped to an (N, T, 1) tensor for LSTM input

[3]. The horizon T is capped to keep sequences comparable across students and to limit very long tails driven by sparse late activity; padding or masking strategies align shorter histories to the same tensor shape for batch training.

4.3 Modeling Strategy

Random Forest branch: categorical encoding, scaling, class-imbalance treatment with SMOTE, and hyperparameter search [2], [11], [12], [13]. Random Forests handle mixed feature types well and yield importances that can be inspected for sanity checks (e.g., confirming that assessment signals participate meaningfully). SMOTE mitigates imbalance by synthesizing minority examples; while not a substitute for real student diversity, it reduces the tendency to predict the majority class uncritically in moderate imbalance regimes [12].

LSTM branch: weekly sequence learning with early stopping to control overfitting and improve generalization [3], [14], [15]. Architectures typically include one or more LSTM layers followed by dense layers with dropout; exact depth and width are selected with validation monitoring. Early stopping uses held-out performance curves so that training halts when temporal patterns cease to generalize.

Hybrid branch: weighted fusion of predicted probabilities to improve robustness across different student behavior profiles.

4.4 Risk Mapping

Hybrid probabilities are translated into intervention-friendly risk levels:

1. High risk: `p >= 0.7`
2. Medium risk: `0.4 <= p < 0.7`
3. Low risk: `p < 0.4`

This mapping converts model outputs into categories that are easier for advisors and support teams to act on.

Thresholds should be treated as institutional parameters: a campus prioritizing aggressive outreach may lower the high-risk cut, while a campus with limited advising capacity may raise it to concentrate resources [16], [17].

4.5 Evaluation Considerations

Model assessment should combine discrimination metrics with confusion-based views. Receiver operating characteristic (ROC) analysis and precision–recall behavior remain relevant when costs of false negatives and false positives differ [16], [17]. In early warning, false negatives (missed at-risk students) are often especially costly; therefore, reporting recall for the non-success class alongside overall accuracy helps align technical evaluation with advising goals. The pipeline includes visual diagnostics such as class distribution and confusion matrices to make trade-offs explicit rather than hiding them behind a single scalar score.

4.6 Experimental Protocol, Splits, and Reproducibility

To obtain honest estimates of generalization, student records are partitioned into training and held-out evaluation sets using a randomized split at the same granularity as the prediction unit (student–module presentation), rather than splitting isolated rows that might belong to the same student across weeks. When students appear in multiple presentations, care is taken so that related records do not straddle train and test in ways that would inflate scores through indirect duplication of individual trajectories. Stratification by outcome label helps preserve class proportions in each split under moderate imbalance, which stabilizes both the Random Forest with SMOTE and the early-stopping criterion for the LSTM.

The Random Forest branch applies SMOTE on training folds only; the validation and test portions remain untouched so that synthetic oversampling does not leak into performance estimation [12].

Hyperparameter search for the forest uses randomized search over bounded ranges [11], while the LSTM uses validation loss for early stopping with patience and dropout for regularization [14], [15]. Random seeds are fixed for stochastic components where feasible so that experiments can be rerun with comparable trajectories; minor numerical drift may still occur across hardware and library versions, so exact bitwise reproduction is not claimed, but the pipeline is structured to minimize unnecessary nondeterminism.

Software stack. Implementations follow common scientific Python practice: tabular preprocessing and Random Forest training use scikit-learn-compatible workflows [13], while sequence modeling uses a deep learning framework with Keras-style or PyTorch-style training loops [14], [15]. Reporting and interactive review can be hosted in lightweight dashboard tools [18].

Together, these choices prioritize transparency and transferability to institutional data science teams more than the novelty of the framework.

4.7 Summary of Engineered Signals

Table 1 consolidates the engineered features used in the hybrid pipeline. The intent is to make the modeling inputs auditable: each column corresponds to a concrete aggregation rule derived from OULAD tables, which supports later explanation and institutional review.

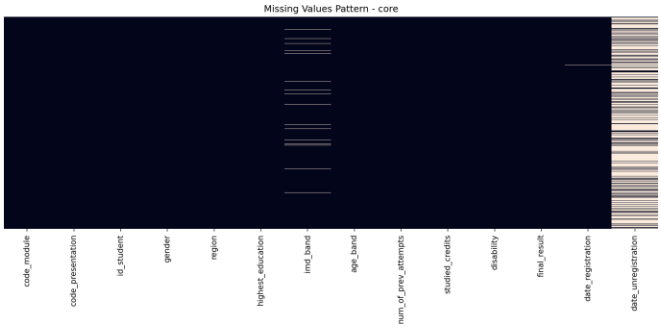
Table 1. Overview of engineered features by branch.

Branch	Feature Group	Representative Inputs	Role
Tabular (RF)	Assessment	Count of submitted assessments, mean score, mean delay	Academic standing and pacing
Tabular (RF)	VLE aggregate	Total clicks, active days, average clicks per active day	Intensity and regularity of engagement
Sequential (LSTM)	Weekly VLE	Click counts per calendar week (bounded horizon)	Temporal trajectory and momentum
Hybrid output	Fused score	$0.6 \cdot p_{RF} + 0.4 \cdot p_{LSTM}$	Combined evidence for risk mapping

5. Essential EDA (Required Visual Evidence)

This section summarizes exploratory findings that shaped preprocessing and model design. To keep the paper readable, only one representative figure is included in this subsection, while complementary evidence is explained in text.

5.1 Missingness and Data Quality



The missingness pattern confirms that incompleteness is concentrated in a limited set of variables rather than being uniformly distributed. This informed a targeted cleaning strategy: essential identifiers were strictly validated, sparsely populated fields with little predictive value were excluded, and missing numeric values in retained features were imputed using stable summary statistics. This approach reduced noise while preserving the behavioral signal needed for early prediction.

5.2 Feature Correlation Structure

Correlation inspection showed that engagement metrics are related but not redundant. For example, total clicks and active days move together, yet each captures a different aspect of participation intensity. Similarly, assessment delay and average score are associated but reflect distinct behavioral patterns. These findings supported the decision to use non-linear learners that can capture interactions without requiring heavy manual feature crossing [2], [8]. High correlation also cautions against interpreting multiple collinear features as independent "causes" in isolation; importance measures should be read as contributions within a joint model.

5.3 Distributional Separation by Outcome

Class-wise distribution checks indicated meaningful separation between successful and non-successful groups across both background and activity-derived attributes. Students in the non-success class generally showed lower sustained platform activity and less consistent assessment behavior over time. This reinforced the hybrid design choice: aggregate features contribute an interpretable baseline context, while sequential inputs capture evolving engagement trajectories.

5.4 Temporal Activity Patterns

Beyond static histograms, reviewing week-by-week activity profiles highlighted common risk signatures: bursts of clicks near deadlines without sustained engagement, or declining weekly counts after an initial spike. Such patterns motivate the LSTM branch specifically, because they are not always visible when only totals and averages are stored. This qualitative inspection also supports transparency: advisors can relate model behavior to recognizable study habits rather than purely abstract scores.

5.5 Base Rates, Priors, and Decision Context

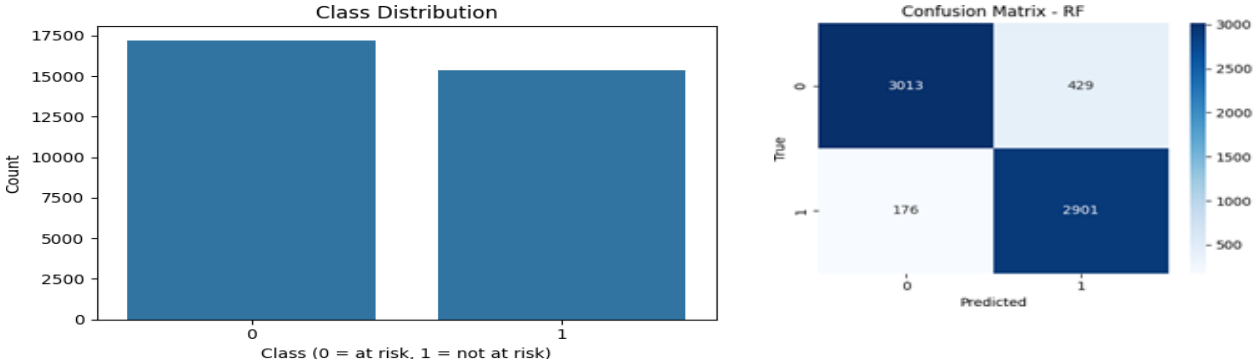
An exploratory review of outcome prevalence reinforces that predictive systems operate against a background rate of non-success that is neither negligible nor extreme in many module presentations. In such settings, even models with respectable accuracy can be dominated by majority-class behavior unless metrics and thresholds are chosen carefully [16], [17]. From an intervention standpoint, the more actionable question is often conditional: among students flagged as high risk, what proportion truly needed support, and among students not flagged, how many were missed? These questions align

naturally with precision–recall reasoning and confusion-matrix inspection rather than accuracy alone. The hybrid design does not remove this fundamental tension, but it provides multiple behavioral channels so that single-feature blind spots are less likely to determine the final risk estimate.

6.Experimental Outputs and Results

6.1 Required Model Output Charts

To avoid overloading the narrative with visuals, this section presents only the most decision-relevant plots.



The class distribution plot confirms a moderate imbalance, which justifies the use of SMOTE in the Random Forest branch and careful threshold tuning during evaluation [12]. The confusion matrix gives a clearer operational view than accuracy alone: it shows where the model tends to miss at-risk learners and where it raises precautionary alerts. In an intervention setting, this trade-off is acceptable when recall for the non-success class remains strong.

6.2 Comparative Branch Behavior (Qualitative)

Although this paper emphasizes architecture and reporting over leaderboard-style benchmarking, comparing branch behaviors remains instructive. The Random Forest branch typically leverages assessment-centric signals strongly, aligning with institutional notions of academic standing. The LSTM branch can adjust predictions when weekly trajectories indicate deteriorating engagement, even if cumulative totals appear adequate. Fusion aims to reduce cases where either representation alone is misleading due to noise or cohort heterogeneity [5], [10].

7.Discussion

The system is built around a practical balance between interpretability and temporal sensitivity. Random Forest contributes clear feature-level explanations that stakeholders can interpret, while LSTM captures progression patterns across weeks [2], [3]. Their weighted fusion is straightforward to reproduce and can be tuned without introducing excessive implementation complexity [4], [8].
In early-warning contexts, sensitivity to at-risk students is often more important than maximizing overall accuracy. Missing a student who needs support may be more costly than generating additional follow-up alerts. For this reason, threshold selection and risk-band calibration should be treated as primary design decisions rather than minor post-processing steps [6], [9], [16], [17].

Ethics, privacy, and responsible use. Predictive analytics in education raises concerns about surveillance, stigmatization, and inequitable treatment if models encode historical bias or are used punitively [4], [8]. Risk scores should support human decision-making rather than replace it; transparency about data sources, uncertainty, and known limitations is essential. Any deployment should include governance review, clarity on who sees scores, and pathways for students to contest or clarify alerts.

Operational integration. Connecting model outputs to advising workflows requires more than technical accuracy: staff training, clear escalation rules, and integration with appointment systems determine whether alerts translate into support [6]. Dashboard prototypes and reporting exports (for example, Streamlit-based summaries [18]) can bridge the gap between offline experiments and review meetings.

Monitoring and continuous validation. A deployed early-warning pipeline should be monitored like other production decision systems: outcome rates, alert volumes, and basic calibration checks should be reviewed each term so that drift is detected before it harms students or wastes staff time. Simple monitoring dashboards can track the distribution of predicted probabilities over time, the distribution of risk bands, and the realized prevalence of non-success outcomes among flagged and unflagged groups. When alert rates spike without a corresponding change in student needs, thresholds, or fusion weights may require re-tuning; when the opposite occurs—few alerts but rising failures—recall may have collapsed due to upstream data changes (for example, logging policy changes in the LMS). These operational checks complement offline metrics and keep the hybrid model aligned with institutional reality [4], [8].

The current study has limitations: it relies on one benchmark dataset, cohort behavior may drift across academic terms, and fairness analysis across demographic groups remains necessary before real deployment [4], [8]. OULAD's public nature aids reproducibility but may not reflect every institution's LMS instrumentation or policy context. Even so, the pipeline demonstrates a workable route from raw LMS traces to intervention-oriented risk estimates [18].

8.Conclusion

This paper introduces a hybrid framework for early student performance estimation using OULAD [1]. By combining tabular aggregate features with sequential clickstream signals, the approach captures complementary information and converts predictions into usable risk categories for academic support teams [2], [3], [4]. The implementation also includes essential EDA outputs, comparative model visuals, and reporting artifacts aligned with practical use.

Future work should explore adaptive fusion weights, cross-cohort external validation, fairness-aware thresholding, and integration with near real-time LMS dashboards. Additional directions include semi-supervised or transfer approaches when labels are scarce in new courses, and student-facing explanations that emphasize actionable study strategies rather than only risk scores.

REFERENCES

[1] J. Kuzilek, M. Hlosta, and Z. Zdrahal, "Open University Learning Analytics Dataset," Scientific Data, 2017. [https://doi.org/10.1038/sdata.2017.171](https://doi.org/10. 1038/sdata.2017.171)

[2] L.Breiman,"RandoForests,"MachineLearning,2001. [https://doi.org/10.1023/A:1010933404324](https://doi.org/ 10.1023/A:1010933404324)

[3] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, 1997. [https://doi.org/10.1162/neco.1997.9.8.1735](https://doi.org /10.1162/neco.1997.9.8.1735)

[4] C. Romero and S. Ventura, "Educational Data Mining and Learning Analytics: An Updated Survey," WIREs Data Mining and Knowledge Discovery, 2020. [https://doi.org/10.1002/widm.1355](https://doi.org/10.100 2/widm.1355)

[5] N. Thai-Nghe, L. Drumond, T. Horvath, A. Krohn-Grimberghe, and L. Schmidt-Thieme, "Matrix and Tensor Factorization for Predicting Student Performance," in Educational Data Mining, 2011. https://educationaldatamining.org/EDM2011/wp-content/uploads/proc/edm11_paper15.pdf

[6] C. Macfadyen and S. Dawson, "Mining LMS Data to Develop an 'Early Warning System' for Educators," Computers&Education,2010. [https://doi.org/10.1016/j.compedu.2010.04.014](https://doi

.org/10.1016/j.compedu.2010.04.014)

[7] R. S. Baker and K. Yacef, "The State of Educational Data Mining in 2009: A Review and Future Visions," JEDM, 2009. https://jedm.educationaldatamining.org/index.php/JEDM/article/view/8

[8] J. L. Rastrollo-Guerrero, J. A. Gómez-Pulido, and A. Durán-Domínguez, "Analyzing and Predicting Students' Performance by Means of Machine Learning: A Review," Applied Sciences, 2020. https://doi.org/10.3390/app10031047

[9] M. V. López, M. C. Burgos, and J. M. C. Moreno, "Using Learning Analytics to Predict Students Performance in Moodle LMS," IEEE EDUCON, 2018. https://doi.org/10.1109/EDUCON.2018.8363288

[10] S. Gray, M. McGuinness, and P. Owende, "An Application of Classification Models to Predict Learner Progression in Third-Level Education," Education and Information Technologies, 2014. https://doi.org/10.1007/s10639-013-9293-y

[11] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," JMLR, 2012. https://jmlr.org/papers/v13/bergstra12a.html](https://jmlr.org/papers/v13/bergstra12a.html)

[12] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," JAIR, 2002. https://doi.org/10.1613/jair.953

[13] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," JMLR, 2011. https://jmlr.org/papers/v12/pedregosa11a.html

[14] F. Chollet et al., "Keras," 2015. https://keras.io/

[15] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," NeurIPS, 2019. https://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library

[16] T. Fawcett, "An Introduction to ROC Analysis," Pattern Recognition Letters, 2006. https://doi.org/10.1016/j.patrec.2005.10.010

[17] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves," in ICML, 2006. https://dl.acm.org/doi/10.1145/1143844.1143874

[18] Streamlit, "Streamlit Documentation," 2024. https://docs.streamlit.io/