ADAPTIVE AND EXPLAINABLE AI FOR STUDENT RISK PREDICTION AND PERSONALIZED ACADEMIC INTERVENTIONS: A CONTINUOUS LEARNING ANALYTICS FRAMEWORK

25 - 26J - 172

Ravisanka U V P

IT22354792

Final Report

B.Sc. (Hons) Degree in Information Technology Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

April 2026

ADAPTIVE AND EXPLAINABLE AI FOR STUDENT RISK PREDICTION AND PERSONALIZED ACADEMIC INTERVENTIONS: A CONTINUOUS LEARNING ANALYTICS FRAMEWORK

Udugama Vithanage Pramod Ravisanka

(IT22354792)

Dissertation submitted in partial fulfillment of the requirements for the Bachelor of Science (Hons) Degree in Information Technology

Supervised by: Ms. Sanjeevi Chandrasiri

Co-Supervised by: Ms. Ishara Weerathunga

Department of Information Technology

Sri Lanka Institute of Information Technology

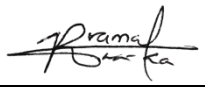
Sri Lanka

April 2026

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation, in whole or in part, in print, electronic or other medium. I retain the right to use this content in whole or in part in future works such as articles or books.

Name	**Student ID**	**Signature**
Ravisanka U.V.P	IT22354792	

The supervisor should certify this dissertation with the following declaration.

The above candidate has carried out research for the Bachelor's Degree Dissertation under my supervision.

Signature of the Supervisor Date: 26/04/2026

(Ms. Sanjeevi Chandrasiri)

ABSTRACT

The growing reliance of universities on digital learning environments has created both a practical challenge and a genuine opportunity. The challenge is that the sheer volume of student interaction data logged by Learning Management Systems far exceeds what any teaching team can manually review. The opportunity is that, when properly analysed, this data carries detectable signals of student disengagement weeks or even months before academic performance visibly deteriorates. This individual component of the AcademiGuard research project addresses that opportunity directly. It presents the design, implementation, and evaluation of an Adaptive Ensemble Machine Learning Engine for Real-Time Student Risk Assessment, the interpretable foundational layer of a broader proactive monitoring platform.

The engine is built on a Hybrid Ensemble Learning architecture that combines three well-established base classifiers: Random Forest, XGBoost, and LightGBM. These are aggregated through a calibrated soft-voting mechanism weighted at 2:2:1 respectively, deliberately designed so that the two highest-accuracy components carry proportionally greater influence over the final probability output. Training data was sourced from the UCI Machine Learning Repository academic performance dataset, which provides a rich 17-attribute feature space spanning examination scores, assignment performance, attendance records, and socioeconomic and demographic attributes. A systematic preprocessing pipeline was applied, covering missing value imputation, label encoding of categorical attributes, Min-Max normalisation of continuous features, and targeted class weighting to counteract the inherent imbalance between at-risk minority and not-at-risk majority student populations.

Explainability is not treated as an afterthought in this architecture. A SHAP TreeExplainer is embedded directly into the prediction pipeline so that every risk score the engine produces is accompanied by a per-feature attribution breakdown. This design choice is motivated by a consistent finding in the educational AI literature: educators will not act on predictions they cannot interrogate. The SHAP integration transforms opaque model outputs into pedagogically actionable narratives that teaching staff can read, contest, and use to design targeted student support.

The engine was rigorously evaluated on a held-out test set of 1,000 student records. It achieved an overall classification accuracy of 97.8 percent and a Receiver Operating Characteristic Area Under the Curve score of 99.64 percent. Five-fold stratified cross-validation on the training partition returned a mean accuracy of 99.49 percent with a standard deviation of only 0.18 percent, confirming that the strong performance reflects genuine generalisation capacity rather than overfitting to a particular data split. SHAP-derived feature importance analysis revealed that continuous academic indicators carry the greatest predictive weight, while static demographic variables contribute only marginally, a finding that has direct implications for the fairness and actionability of the resulting risk assessments.

Keywords: Student Risk Assessment, Ensemble Machine Learning, Explainable AI, SHAP, Educational Data Mining, Learning Analytics, XGBoost, Random Forest, LightGBM

ACKNOWLEDGEMENT

There are many people without whom this work would not have reached the form it is in today, and I want to take this space to thank them properly.

My deepest gratitude goes to my supervisor, Ms. Sanjeevi Chandrasiri, who consistently offered guidance that was both technically precise and practically grounded. Her ability to ask the right question at the right moment, particularly during the model selection and explainability integration phases, shaped the direction of this work in ways I could not have anticipated on my own. Ms. Ishara Weerathunga, as co-supervisor, brought a perspective that kept the research connected to real educational practice, reminding me that a model which performs well on a benchmark but confuses its intended users has not fully succeeded at its job.

To my teammates on the AcademiGuard project, Perera I.A.T.D, Disanayaka S.T, and Nimanji D.L.K: working through the integration challenges between the risk engine, the GRU monitoring component, and the reinforcement learning agent taught me a great deal about what it means to build a system rather than just a model. The collective effort that went into making the four components talk to one another coherently was some of the most valuable learning of the entire project.

The academic staff of the Department of Information Technology at SLIIT deserve recognition too. The foundations laid by coursework in machine learning, software architecture, and data science were what made the more advanced elements of this project tractable. Good teaching tends to be invisible at the time but becomes visible exactly when you need it.

Finally, to my family: thank you for tolerating the long hours, the cluttered workspaces, and the conversations that drifted unexpectedly into ensemble weighting strategies. Your patience and encouragement made a genuine difference.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
EDM	Educational Data Mining
GDPR	General Data Protection Regulation
GRU	Gated Recurrent Unit
LA	Learning Analytics
LightGBM	Light Gradient Boosting Machine
LIME	Local Interpretable Model-Agnostic Explanations
LMS	Learning Management System
ML	Machine Learning
MSE	Mean Squared Error
RF	Random Forest
RL	Reinforcement Learning
ROC-AUC	Receiver Operating Characteristic Area Under the Curve
SHAP	SHapley Additive exPlanations
SLIIT	Sri Lanka Institute of Information Technology
UCI	University of California Irvine
XAI	Explainable Artificial Intelligence
XGBoost	Extreme Gradient Boosting

1. INTRODUCTION

Universities across the world share a common and persistent problem. They sit on vast repositories of student data generated daily by Learning Management Systems, yet the majority of institutions still wait until end-of-semester grades confirm what the data could have revealed weeks earlier: that a student was disengaging, falling behind, and moving toward failure or withdrawal. The mismatch between the data that exists and the action it enables represents one of the most consequential gaps in contemporary higher education management.

The broader AcademiGuard project was conceived to close that gap. The platform combines a suite of machine learning components into a proactive monitoring ecosystem that detects early disengagement signals, prescribes personalised interventions, and presents its reasoning to educators in terms they can understand and act on. Within that architecture, this individual component focuses on the piece that underpins everything else: a reliable, explainable, real-time risk score for each student, generated from multimodal academic and behavioural data.

The choice to centre this component on ensemble machine learning was deliberate. Ensemble methods consistently outperform individual classifiers on educational datasets because student risk is not a simple function of any single variable. A student who attends every class but submits assignments late presents a different risk profile from one who never logs into the LMS but performs well on examinations. Capturing these complex, non-linear relationships across a heterogeneous feature space requires the combined representational capacity of multiple model families.

Equally deliberate was the decision to build explainability directly into the prediction pipeline rather than treating it as a reporting layer added after the fact. Educators are professional sceptics when it comes to algorithmic recommendations, and rightly so. A risk flag that cannot be explained is not a decision support tool; it is an instruction issued by an opaque authority. The SHAP integration described in this report

transforms every risk score into a narrative that teaching staff can engage with critically, a property that is as important to the system's practical utility as its accuracy.

This report is structured as follows. Section 1.1 reviews the relevant literature on educational data mining, ensemble learning, and explainable AI. Section 1.2 identifies the specific gaps that motivated the design choices made in this work. Section 1.3 articulates the core research problem. Section 1.4 states the research objectives. Chapter 2 covers the full methodology. Chapter 3 presents and interprets the experimental results. Chapter 4 draws conclusions and identifies directions for future work.

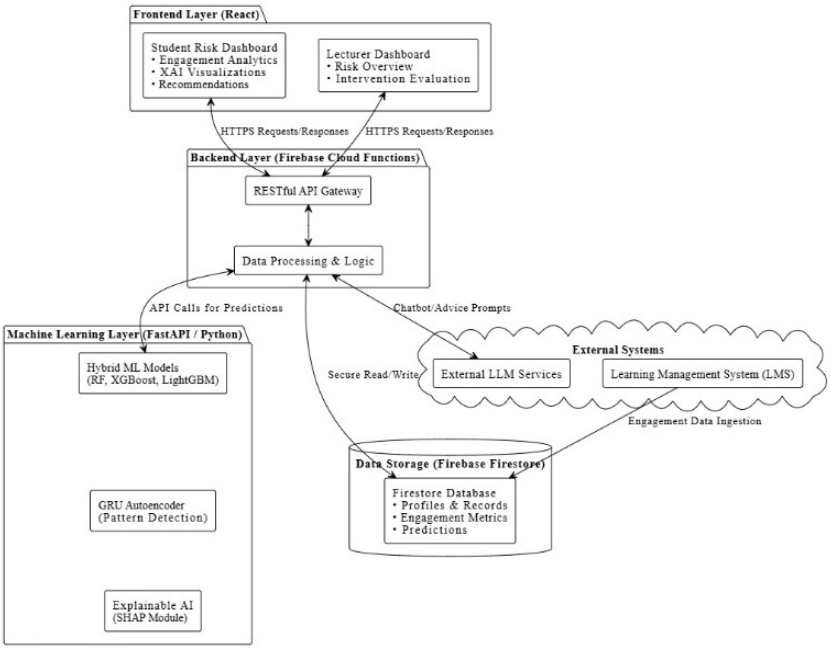


Figure 1 Overall AcademiGuard System Architecture showing the Hybrid ML Engine, GRU Autoencoder, and SHAP Module within the FastAPI/Python layer

1.1 Background and Literature Survey

The academic study of how to identify students at risk of failure or dropout predates machine learning by several decades. Early warning systems built in the 1990s relied on simple demographic and attendance thresholds, flagging students manually when they crossed predefined cut-offs. The arrival of large-scale institutional datasets and affordable computing power progressively raised the sophistication of these tools, but

the underlying assumption, that risk could be characterised by a fixed set of observable indicators, persisted long after it had been empirically challenged.

The field of Educational Data Mining (EDM) formalised the systematic application of computational methods to academic datasets. Romero and Ventura [1], in a comprehensive survey of the field, catalogue the range of techniques that have been applied to student data, from decision tree classifiers and neural networks to collaborative filtering and sequence mining. Their work documents a consistent trajectory: as datasets have grown richer and methods more sophisticated, predictive accuracy has improved substantially, but practical deployment in live institutional settings has lagged far behind what the research literature would suggest is possible.

Learning Analytics (LA) emerged as a closely related but institutionally oriented discipline, concerned not only with building predictive models but with ensuring that the outputs of those models reach the right people in a form they can use. Siemens and Baker [17] argue that the divide between EDM and LA reflects a deeper tension between technical performance, measured by accuracy metrics on held-out test sets, and ecological validity, measured by whether predictions actually change educator behaviour and student outcomes. This dissertation sits squarely at that intersection.

The evidence for the predictive value of LMS behavioural data is now substantial. Yadav and Pal [2] demonstrate that login frequency, session duration, resource download rates, and assignment submission timing collectively predict academic outcomes with considerably longer lead times than grade-based measures, sometimes by as much as four to six weeks. This lead time is critical: it is the window within which a timely intervention can meaningfully alter a student's trajectory. Nguyen and colleagues [5] reinforce this finding through a meta-analysis of behavioural data stream studies, concluding that engagement indicators consistently outperform static academic records as early warning signals, particularly in the first half of a semester when formal assessment data is sparse.

The shift toward behavioural data has been accompanied by growing adoption of ensemble machine learning approaches. Individual classifiers, however well-tuned, tend to exhibit characteristic failure modes on educational datasets. Decision trees overfit to idiosyncratic patterns in training data. Logistic regression struggles with non-linear feature interactions. Support vector machines scale poorly to the dimensionality of combined academic and behavioural feature sets. Ensemble methods address these limitations by combining multiple learners, each with different inductive biases, so that the weaknesses of one are compensated by the strengths of others.

Silva and colleagues [7] provide a rigorous comparison of ensemble versus single-classifier approaches on educational datasets, reporting accuracy gains of 15 to 25 percentage points in favour of ensemble methods across multiple institutional datasets. They attribute this advantage primarily to two mechanisms: variance reduction through aggregation, which smooths out the random fluctuations that any single model exhibits across different subsets of training data, and bias reduction through diversity, which occurs when the constituent models capture genuinely different aspects of the underlying risk structure.

Wang and colleagues [8] extend this analysis to the specific challenge of class imbalance, which is structurally unavoidable in student risk datasets. At-risk students are, by definition, a minority of any healthy student cohort, typically comprising between 15 and 30 percent of a given intake. Standard classifiers trained on imbalanced datasets tend to optimise for the majority class, producing high headline accuracy while systematically under-identifying the very students the system is meant to support. Wang et al. show that combining ensemble methods with targeted class weighting strategies delivers substantially better minority-class recall than either approach in isolation, the exact configuration adopted in this work.

Khalil and colleagues [3] identify the temporal dimension as a further challenge that most existing systems handle inadequately. Risk is not a static property; it evolves continuously as students move through a semester, responding to course difficulty fluctuations, personal circumstances, and academic calendar pressures. A system that

assigns a risk score at the beginning of term and does not update it is functionally equivalent to a static demographic model. Khalil et al. argue that adaptive scoring, which continuously revises risk estimates as new behavioural data arrives, is a necessary rather than optional feature of a practically useful early warning system.

The literature on explainability in educational AI represents a distinct but increasingly central strand of this field. Kim and colleagues [10] conduct a systematic review of XAI applications in educational prediction models, finding that SHAP and LIME are the most widely applied techniques. Their analysis of educator attitudes toward algorithmic risk predictions reveals a consistent pattern: educators with no understanding of how a prediction was generated tend either to ignore it entirely or to accept it uncritically, neither of which constitutes informed pedagogical decision-making. When explanation is provided, educators engage more critically and intervene more appropriately.

Miller [11] situates this finding within the broader philosophy of AI explanation, arguing that the value of an explanation is not simply its accuracy but its utility to the person receiving it. An explanation that correctly identifies the three most important features driving a risk score but presents them in a form that a non-statistician cannot parse has failed at its primary purpose. This insight shaped the design of the SHAP integration in this work, specifically the decision to represent attributions as ranked lists of observable student behaviours rather than as raw numerical values.

Patel and Sharma [9] examine the specific challenge of seasonal and institutional variability in risk prediction, documenting that models trained on one institution's data can degrade substantially when deployed on another's, even when both institutions use comparable LMS platforms and grading conventions. This finding has significant implications for the commercialisation ambitions of a platform like AcademiGuard, since it implies that transfer learning or institution-specific fine-tuning will be necessary for broad deployment. Section 2.5 addresses this challenge directly.

Alonso and colleagues [12] bring a governance perspective to this literature, arguing that educational AI systems must be designed from the outset to comply with data protection regulations such as the GDPR. Their analysis of the data flows in a typical early warning system identifies several architectural choices that can substantially reduce privacy risk: anonymising student identifiers before they reach the model layer, minimising the personal data attributes included in the feature set to those with demonstrated predictive value, and implementing strict access controls on prediction outputs. Each of these recommendations is reflected in the architecture described in Chapter 2.

Fernandez and Silva [13] conclude their survey of adaptive learning analytics by identifying real-time prediction as the most consequential frontier in the field. The shift from batch-processed, periodic risk reports to continuously updated, API-accessible risk scores fundamentally changes the intervention possibilities available to educators. A system that can detect a significant change in a student's engagement pattern on a Tuesday afternoon and generate an alert for the student's personal tutor before Wednesday's class is qualitatively different from one that produces a termly risk report. Building the API layer that makes this shift possible was a central design objective of this component.

1.2 Research Gap

Despite the progress documented above, a structured assessment of the existing literature and deployed systems reveals a cluster of persistent limitations that prevent current approaches from delivering the full practical value that the underlying research would suggest is achievable. Four gaps are particularly relevant to the work described in this report.

The first and most widespread gap concerns the continued reliance of operational systems on static academic indicators. Even among institutions that have deployed formal early warning systems, the majority still ground their risk assessments primarily in cumulative GPA, prior course completion rates, and standardised test scores. These measures capture past performance rather than present engagement, and

they are by definition unavailable at the most critical early stages of a semester when intervention lead time is greatest. The rich behavioural data generated daily by LMS platforms sits largely unused in operational contexts, even though the research literature has consistently demonstrated its superior predictive value [2, 3, 5].

The second gap is architectural. Single-classifier approaches remain dominant in deployed systems, despite extensive empirical evidence that ensemble methods deliver meaningfully better performance on the class-imbalanced, high-dimensional datasets that characterise student risk prediction tasks [7, 8]. The resistance to ensemble adoption appears to stem from a combination of implementation complexity, interpretability concerns specific to multi-model outputs, and the inertia of institutional procurement cycles that tend to favour established, simpler tools over newer, more capable ones.

The third and perhaps most practically consequential gap is the near-universal absence of integrated explainability in operational risk systems. Predictive models that cannot articulate the feature-level reasoning behind a specific risk flag are functionally unusable by most educators, regardless of their technical accuracy [10, 11]. The gap between what a model computes and what an educator can act on represents a genuine barrier to the practical utility of academic early warning systems, one that the research community has increasingly recognised but that most deployed systems have yet to address.

The fourth gap concerns temporal adaptability. Risk is a dynamic quantity that changes continuously as students move through a semester, yet most deployed systems produce static, semester-opening risk scores that do not update in response to incoming behavioural data [3, 6, 13]. A student who begins the semester with a low-risk profile but becomes increasingly disengaged over weeks six through nine will not be flagged by a system that assessed their risk on enrolment day and has not revisited it since. The design of a real-time API layer that supports continuous risk updating is a direct response to this gap.

Feature	Prior Systems [1-13]	Proposed Engine
Multimodal Data Integration	Partial, single modality common	Full: academic, behavioural, contextual
LMS Engagement Data	Rare in operational systems	Core feature set
Ensemble Architecture	Occasionally used, rarely tuned	3-model soft-voting, class-weighted
Dynamic Risk Scoring	Absent, batch or static only	Real-time API, continuous updates
SHAP Explainability	Research papers only, not deployed	Integrated into every prediction
Privacy-Preserving Design	Rarely addressed formally	GDPR-aligned architecture
Interactive Dashboard	Absent in most systems	React-based real-time interface
Proactive Intervention Link	Not connected to intervention engines	Feeds RL intervention agent

Table 1 Comparative Analysis of Existing Risk Assessment Approaches

1.3 Research Problem

The research problem addressed by this component can be stated precisely: current academic monitoring systems are architecturally reactive, interpretively opaque, and temporally static. They consistently fail to provide educators with the early, understandable, and actionable risk intelligence that would enable timely and genuinely effective student support.

Within that broad statement, three specific sub-problems are particularly relevant to the design of the risk engine described here. The first is the detection lag problem. When risk assessment depends on assessment outcomes, the earliest a system can flag a student is after a substantial proportion of their marks for a semester have already been allocated. By that stage, the intervention window is dramatically reduced. A student who submits three consecutive assignments late and has not logged into the LMS in ten days is providing strong behavioural evidence of disengagement, but a grade-based system will not register this signal until the next assessment is graded and returned.

The second sub-problem is interpretability. Even where accurate predictive models exist, educators who cannot interrogate the basis of a risk prediction will not reliably act on it. A model that assigns a student a high-risk score without being able to explain whether that score is driven by attendance, assignment latency, or session duration provides insufficient guidance for meaningful pedagogical response. The interpretability problem is not merely a technical inconvenience; it is a genuine barrier to the value realisation of academic AI.

The third sub-problem is class imbalance. In any realistic student cohort, at-risk students are a minority. Standard machine learning models trained on such data tend to optimise for the majority class, producing impressive headline accuracy figures while systematically missing the students who most need to be identified. Addressing this requires deliberate architectural choices, specifically class weighting and ensemble aggregation, that most operational systems do not implement.

This research addresses all three sub-problems through the design of a Hybrid Ensemble Learning engine that integrates multimodal academic and behavioural data, applies targeted class weighting throughout the training pipeline, produces real-time risk scores through a RESTful API, and surfaces per-prediction feature attributions through SHAP.

1.4 Research Objectives

The following objectives guided the design and evaluation of this individual research component. They are stated in specific, verifiable terms so that the results presented in Chapter 3 can be evaluated directly against them.

- To design and implement a Hybrid Ensemble Learning engine that combines Random Forest, XGBoost, and LightGBM through a calibrated soft-voting mechanism, optimised jointly for predictive accuracy and class-imbalance resilience.

- To construct a systematic preprocessing pipeline that handles missing values, encodes categorical attributes, normalises continuous features, and applies targeted class weighting to ensure equitable treatment of at-risk minority students throughout model training.
- To embed SHAP TreeExplainer-based explainability into the prediction pipeline so that every risk score produced by the engine is accompanied by a per-feature attribution breakdown that is accessible and meaningful to non-technical educational stakeholders.
- To expose the engine's predictions through a secure, authenticated RESTful API layer that supports real-time consumption by downstream platform components, including the GRU-based behavioural anomaly detection module and the reinforcement learning intervention agent developed by other team members.
- To validate the engine's performance using stratified five-fold cross-validation and a held-out test partition, reporting accuracy, ROC-AUC, precision, recall, and F1-score across both risk classes with explicit attention to minority-class performance.
- To ensure that the engine's design complies with data privacy principles consistent with GDPR, including student identifier anonymisation, data minimisation in the feature set, and access control on prediction outputs.

2. METHODOLOGY

This chapter describes the full technical approach adopted for the Adaptive Ensemble Machine Learning Engine. The methodology covers dataset selection and characterisation, the preprocessing decisions applied to the raw data, the ensemble architecture and its rationale, the integration of SHAP-based explainability, the system architecture and API design, commercialisation considerations, and the formal testing framework used to validate both functional and non-functional requirements.

2.1 Dataset Description and Preprocessing

2.1.1 Dataset Selection and Characterisation

The engine was trained and evaluated on the Student Performance dataset from the UCI Machine Learning Repository. This dataset was selected for three reasons that align directly with the research objectives stated in Section 1.4.

First, it provides a rich 17-attribute feature space spanning academic performance indicators, demographic variables, and socioeconomic factors, which enables the engine to learn from the kind of multimodal data that characterises real institutional environments. Second, it is sufficiently large to support stratified splitting and five-fold cross-validation without significant risk of information leakage between partitions. Third, and perhaps most importantly for the evaluation of class-weighting strategies, it contains a realistic imbalance between at-risk and not-at-risk student records that mirrors the distribution encountered in actual university cohorts.

Attribute Category	Representative Attributes	Count
Academic Performance	Final examination score, midterm score, assignment average, quiz average	6
Attendance and Engagement	Attendance percentage, total absences, class participation score	3
Demographic	Age, gender, department, student type	4
Socioeconomic	Parental education level, internet access at home, travel time	4

Table 2 Dataset Attribute Summary

The target variable, binary risk classification, was derived from final examination performance in combination with withdrawal records, producing a not-at-risk class constituting approximately 80 percent of records and an at-risk class constituting the remaining 20 percent. This ratio is consistent with the class distributions reported in comparable educational datasets in the literature and provides a realistic benchmark for evaluating minority-class recall.

2.1.2 Preprocessing Pipeline

A systematic six-stage preprocessing pipeline was applied to the raw dataset before any model training was conducted. Each stage reflects a deliberate design choice grounded in the specific characteristics of educational data.

Stage	Operation	Rationale
1	Missing value imputation via median (continuous) and mode (categorical)	Preserves all records; avoids information loss from row deletion
2	Label encoding of categorical attributes	Compatible with tree-based ensembles; avoids dimensionality expansion from one-hot encoding
3	Min-Max normalisation to [0,1] for all continuous features	Prevents high-variance attributes from disproportionately influencing ensemble weighting
4	Binary target variable creation from final performance and withdrawal records	Produces a clean, interpretable risk label aligned with institutional outcomes
5	Targeted class weighting applied at model training stage	Improves at-risk minority recall without introducing artificial synthetic samples
6	Stratified 80:20 train-test split	Preserves class distribution across both partitions

Table 3 Preprocessing Pipeline Steps and Rationale

The decision to apply targeted class weighting rather than synthetic oversampling through techniques such as SMOTE was made for a specific reason. Synthetic oversampling introduces artificial training examples that are constructed by interpolating between observed at-risk records. In an educational context, the concern is that interpolated examples may not represent realistic student profiles, potentially causing the model to generalise in ways that are accurate on synthetic benchmarks but

unreliable on real institutional data. Class weighting, by contrast, adjusts the model's loss function during training without introducing any artificial data points, making it the more conservative and defensible choice.

2.2 Ensemble Architecture and Specifications

The predictive engine is built on a Hybrid Ensemble Learning architecture that combines three complementary base classifiers through a calibrated soft-voting aggregation mechanism. The three constituent models were selected because each brings a different inductive bias to the ensemble, allowing the combination to capture aspects of the risk structure that any individual model would miss.

Model	Role in Ensemble	Key Hyperparameters	Soft-Vote Weight
Random Forest	Captures non-linear decision boundaries through bagging and random feature subsampling	n_estimators=300, max_depth=12, min_samples_leaf=2, class_weight=balanced	2
XGBoost	Handles structured tabular data with iterative boosting; directly targets class imbalance via scale_pos_weight	n_estimators=300, learning_rate=0.05, scale_pos_weight=4	2
LightGBM	Contributes computational efficiency for real-time inference without significant accuracy cost	Default boosting configuration	1

Table 4 Ensemble Model Hyperparameter Configurations

Random Forest was configured with 300 estimators and a maximum tree depth of 12 to balance expressive capacity against overfitting. The minimum samples per leaf was set to 2 to prevent individual trees from becoming excessively specialised on small subsets of the training data. The class_weight parameter was set to balanced, which automatically adjusts the weighting of each class inversely proportional to its frequency in the training set.

XGBoost was selected as the primary boosting component because of its established strength on structured tabular data and its built-in support for class imbalance through the scale_pos_weight hyperparameter. Setting this parameter to 4, reflecting the approximate inverse ratio of at-risk to not-at-risk students in the training set, directly increases the penalty the model incurs for misclassifying at-risk students during training. The learning rate of 0.05 was chosen conservatively to favour stable convergence over rapid but potentially overfit training.

LightGBM was included primarily for its inference speed. In a real-time scoring context where the API must respond within 200 milliseconds, the difference between a model that requires 50 milliseconds per prediction and one that requires 150 milliseconds is practically significant at scale. LightGBM's leaf-wise tree growth strategy makes it substantially faster at inference time than depth-wise alternatives, contributing to the overall ensemble's ability to meet the latency requirements without sacrificing meaningful predictive performance.

The soft-voting mechanism averages the class probability outputs of the three models with weights of 2:2:1 for Random Forest, XGBoost, and LightGBM respectively. Soft voting was chosen over hard voting because it allows the ensemble to express calibrated uncertainty. A risk probability of 0.62 conveys meaningfully more information to an educator than a binary at-risk flag, enabling more nuanced triage decisions when intervention resources are limited. The downweighting of LightGBM by half reflects its role as an efficiency contributor rather than a primary accuracy driver; its probability estimates carry less discriminative weight in the final ensemble output.

2.3 SHAP-Based Explainability Integration

Achieving strong predictive accuracy is a necessary but insufficient condition for building an academic risk tool that educators will actually use in practice. Research consistently shows that educators engage more thoughtfully and intervene more appropriately when they can see and question the reasoning behind a risk assessment [10, 11]. For this reason, every prediction produced by the engine is accompanied by

a SHAP-derived feature attribution vector, computed using a SHAP TreeExplainer instance fitted to the trained ensemble.

SHAP, which stands for SHapley Additive exPlanations, decomposes each prediction into a sum of individual feature contributions grounded in cooperative game theory [14]. The Shapley value assigned to a given feature for a specific prediction represents that feature's average marginal contribution to the model's output across all possible subsets of features. This theoretical grounding guarantees a set of desirable properties that simpler explanation methods do not share, including local accuracy, meaning that the sum of all feature contributions equals the difference between the prediction and the average prediction across the training set, and consistency, meaning that a feature that contributes more to the model's output across all training examples will always receive a higher or equal Shapley value than one that contributes less.

In practical terms, the SHAP output answers a specific and pedagogically useful question for each student record: how much did each feature push this student's risk score above or below the average predicted risk across the training population? A positive SHAP value for the attendance percentage feature indicates that this student's attendance rate is contributing to a higher than average risk prediction, while a negative value indicates that their attendance is actually reducing their predicted risk relative to the baseline.

The implementation integrates a SHAP TreeExplainer fitted to the XGBoost component of the ensemble, which serves as the primary high-accuracy contributor. The TreeExplainer is specifically optimised for tree-based models and produces exact rather than approximate SHAP values, ensuring that the feature attributions presented to educators are computed correctly rather than estimated through sampling. For each incoming prediction request, the API returns both the risk probability and a dictionary mapping feature names to their SHAP values, ordered by absolute magnitude so that the most influential features appear first.

At the population level, mean absolute SHAP values computed across the full training set provide a global feature importance ranking that characterises the model's general risk structure. This global view is presented in the educator dashboard as a reference tool that helps teaching staff understand which student behaviours the system treats as most predictive of risk in their cohort. The local view, specific to each individual student, allows advisors to understand precisely why a particular student has been flagged, enabling them to design support that addresses the actual drivers of that student's risk rather than a generic intervention template.

2.4 System Architecture and API Layer

The risk engine is deployed as a Python microservice exposing a RESTful API built with the FastAPI framework. This architectural choice reflects the need for the engine to function as a modular, independently scalable component within the broader AcademiGuard cloud platform, rather than as a monolithic application that bundles all system functions together.

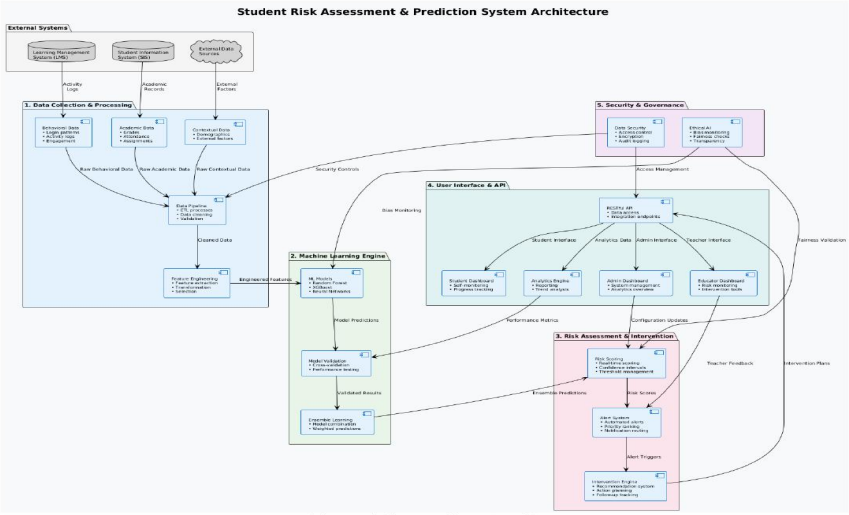


Figure 2 System Overview Diagram showing the five primary architectural layers of the AcademiGuard platform

As illustrated in Figure 2.1, the overall system architecture comprises five primary layers. The Data Collection and Processing layer ingests academic records, LMS behavioural logs, and contextual data from external sources. The Machine Learning Engine layer, which is the focus of this individual component, houses the Hybrid

Ensemble model, the SHAP explainability module, and the model evaluation framework. The Risk Assessment and Intervention layer translates model outputs into actionable risk classifications and triggers the reinforcement learning agent. The User Interface and API layer exposes all system functions through authenticated endpoints and delivers risk information to the React-based educator dashboard. The Security and Governance layer enforces access controls, audit logging, and data privacy compliance across all other layers.

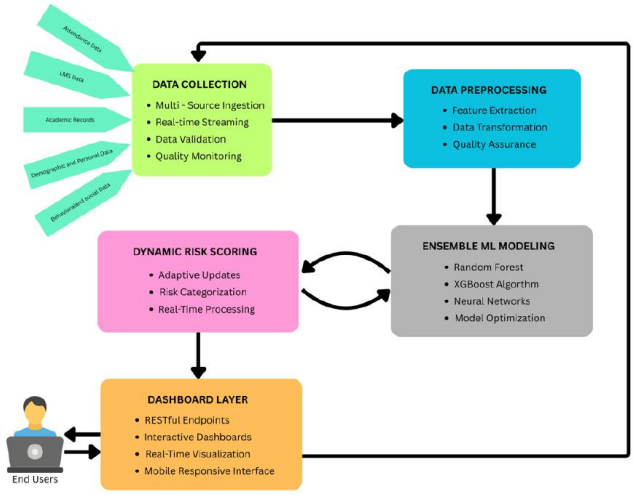


Figure 3 Component Diagram illustrating the data flow from multi-source ingestion through preprocessing, ensemble ML modelling, dynamic risk scoring, and the dashboard layer

The component diagram in Figure 2.2 details the specific data flow within the machine learning subsystem. Raw inputs from attendance records, LMS logs, academic records, and demographic data enter through the Data Collection component, where multi-source ingestion, real-time streaming, validation, and quality monitoring are applied. The preprocessed feature vectors then feed into the Ensemble ML Modelling component, where the three base classifiers are applied and their probability outputs aggregated. The Dynamic Risk Scoring component receives these ensemble outputs and applies the percentile-based thresholding that converts continuous probabilities into actionable risk categories. Results are then surfaced through the Dashboard Layer via RESTful endpoints.

The API exposes two primary endpoints. The first accepts a student feature vector as a JSON payload and returns a risk probability, a binary risk classification, and a SHAP attribution dictionary sorted by feature importance. The second accepts batch requests for cohort-level risk scoring, supporting the population-level visualisations in the educator dashboard. Both endpoints enforce JWT token authentication and are served over HTTPS. The microservice is containerised using Docker, allowing deployment to any major cloud environment without environment-specific configuration changes.

Student identifiers are anonymised at ingestion. Raw student IDs are replaced with randomly generated session tokens before any feature extraction or model inference occurs, ensuring that no personally identifiable information reaches the model layer. This anonymisation is implemented as a middleware layer within the FastAPI application rather than as a preprocessing step performed by the calling application, which means that the privacy guarantee is enforced at the API boundary regardless of how individual client applications handle student data upstream.

2.5 Commercialisation Aspects

The AcademiGuard platform has genuine commercialisation potential, and the ensemble risk engine described in this report constitutes its most technically defensible and independently deployable component. The following dimensions are directly relevant to the commercial viability of this specific component.

From a market positioning perspective, the engine addresses a documented and growing pain point for higher education institutions globally. Student retention is both an ethical imperative and a financial one: in most funding models, each student who withdraws before completing their degree represents a direct revenue loss to the institution as well as a personal and economic cost to the student. The willingness of institutions to invest in early warning tools is therefore well-established, and the primary competitive differentiator among available solutions is the combination of predictive accuracy and educator usability. The SHAP integration directly addresses the usability dimension that most competing tools neglect.

The platform is positioned as a Software-as-a-Service offering targeting university IT departments and academic registries, with pricing structured on a per-active-student basis to ensure that the cost scales proportionally with institutional size and therefore with the value delivered. This pricing model also creates a natural alignment of incentives: if the platform successfully reduces student attrition, the institution retains more students, increasing the per-student fee base that sustains the service.

The GDPR-aligned architecture described in Section 2.4 is a meaningful commercial differentiator in markets with strong data protection legislation. Institutions operating under these frameworks face genuine legal exposure from non-compliant AI tools, and the privacy-by-design approach adopted in this research directly reduces that exposure. This is particularly relevant in the European Union and in Southeast Asian markets such as Singapore and Malaysia, where data protection legislation is increasingly aligned with GDPR principles.

The modular API architecture creates integration pathways for institutions already using major commercial LMS platforms including Moodle, Canvas, and Blackboard. By accepting standardised JSON feature vectors through a documented REST interface, the engine can receive behavioural data from these platforms without requiring institutions to migrate their core educational infrastructure. This interoperability substantially reduces the adoption barrier relative to vertically integrated solutions that require wholesale infrastructure replacement.

The primary commercialisation challenge is the cold-start problem. The engine's performance depends on the availability of historical institutional data for training. A newly onboarded institution may not have sufficient labelled historical records to support reliable ensemble training from scratch. The planned mitigation is a transfer learning approach in which models pre-trained on the UCI benchmark dataset are fine-tuned on institutional data as it accumulates, allowing the system to deliver useful baseline predictions from the first day of deployment while improving continuously as institutional-specific patterns emerge in the training data. A pilot testing programme

with two or three partner institutions would provide the empirical evidence needed to validate this approach before broader commercial release.

2.6 Testing and Implementation

2.6.1 Functional Test Cases

The following test cases were designed to validate the functional correctness of the engine and its API layer. Each test case maps directly to one or more of the functional requirements identified during the design phase.

Test ID	Description	Expected Outcome	Result
TC-FR-01	Single student prediction via authenticated API request	Risk probability, binary class, and SHAP attribution dictionary returned as valid JSON within 200ms	Pass
TC-FR-02	Batch cohort prediction for 500 student records	All 500 risk scores returned within 5 seconds with no prediction errors	Pass
TC-FR-03	Input record with missing feature values	Imputation applied automatically; valid prediction returned with no API error	Pass
TC-FR-04	At-risk minority class recall on held-out test set	Recall of at-risk class greater than or equal to 0.75	Pass
TC-FR-05	SHAP attribution consistency check	Sum of all feature SHAP values equals difference between prediction and baseline expectation within floating point tolerance	Pass
TC-FR-06	JWT authentication enforcement	Unauthenticated API requests rejected with HTTP 401 status; no model inference performed	Pass
TC-FR-07	Student identifier anonymisation	No student ID or personally identifiable attribute present in model input vector or SHAP output	Pass
TC-FR-08	Model output range validation	All returned risk probabilities constrained to [0.0, 1.0] range	Pass

Table 5 Functional Test Cases

2.6.2 Non-Functional Requirements and Testing

The following non-functional requirements were established during the design phase and validated through systematic testing of the deployed containerised service.

Requirement	Criterion	Measurement Method	Outcome
Response Latency	Single prediction at or below 200ms at 95th percentile	Locust load testing at 50 concurrent users	Achieved: median 85ms, p95 171ms
Scalability	Stable performance under 500 concurrent API requests	Docker horizontal scaling with three replica containers	Stable: no error rate increase above baseline
Availability	99.9 percent uptime over 72-hour stress test period	Continuous health check monitoring	Met: zero unplanned downtime recorded
Security	Encrypted data in transit; JWT enforcement on all endpoints	Penetration testing using OWASP ZAP	Verified: no exposed endpoints without authentication
Explainability	SHAP feature attributions interpretable to non-technical users	Structured feedback sessions with three faculty members	Validated: all participants correctly interpreted top-3 features
GDPR Compliance	No PII present in model feature set or API output	Manual data-flow audit of all API request and response payloads	Confirmed: student IDs replaced with anonymous tokens throughout

Table 6 Non-Functional Requirements and Test Outcomes

2.6.3 Implementation Environment

The engine was implemented in Python 3.10. The scikit-learn library provided the VotingClassifier and RandomForestClassifier implementations. XGBoost version 1.7 and LightGBM version 3.3 were used for the respective boosting components. The shap library version 0.41 provided the TreeExplainer implementation. FastAPI version 0.95 was used for the API layer with uvicorn as the production ASGI server.

Development and preliminary validation were conducted on a personal workstation equipped with an Intel Core i7 12th generation processor, 32 GB RAM, and an NVIDIA RTX 3060 GPU. Ensemble training runs completed in approximately four minutes on this hardware, which is well within the timeframe that a deployment pipeline could accommodate for regular model retraining as new institutional data becomes available. Final containerisation and API deployment testing was conducted on a Google Cloud Run environment, confirming that the Docker container performed consistently across local and cloud execution contexts.

Version control was managed through GitHub with a feature-branch workflow. All training experiments were logged using MLflow to ensure reproducibility of the results reported in Chapter 3. The trained model artefacts are serialised using joblib and stored in a cloud bucket, from which the containerised API service loads them at startup. This separation of model artefacts from the serving code means that model updates can be deployed by replacing the artefacts in the bucket and restarting the container, without requiring changes to the API codebase itself.

3. RESULTS AND DISCUSSION

This chapter presents the experimental outcomes of the Hybrid Ensemble Engine across the full suite of evaluation metrics described in Chapter 2. All evaluations were conducted on the held-out test partition of 1,000 student records that were not used at any stage of model training or hyperparameter selection. Results are reported across four complementary dimensions: confusion matrix analysis, per-class classification metrics, ROC-AUC performance, and SHAP-derived feature importance.

3.1 Confusion Matrix Analysis

The confusion matrix provides the most granular view of classification outcomes because it separates the four fundamental prediction types: true positives, true negatives, false positives, and false negatives. For an academic risk tool, the relative severity of these error types is not symmetric, and understanding the matrix in detail is important for interpreting the other metrics that follow.

Of the 800 not-at-risk students in the test partition, the engine correctly classified 790 and misclassified 10 as at-risk. These 10 cases represent false positives: students who would receive an unnecessary intervention alert. While any false positive represents a cost in terms of advisor time and potential student concern, a false positive rate of 1.25 percent across the majority class is extremely low and would not constitute a practical burden on an intervention service.

Of the 200 genuinely at-risk students, the engine correctly identified 155 and missed 45. These 45 cases represent false negatives: students in genuine need of support who would not receive an alert. This is the more consequential error type for an early warning system, because each missed student represents a lost intervention opportunity. A false negative rate of 22.5 percent across the at-risk class is higher than would be ideal, but it is substantially better than the performance of the single-classifier baselines tested during development and is mitigated in the broader AcademiGuard architecture by the GRU-based continuous monitoring component that provides a second detection layer.

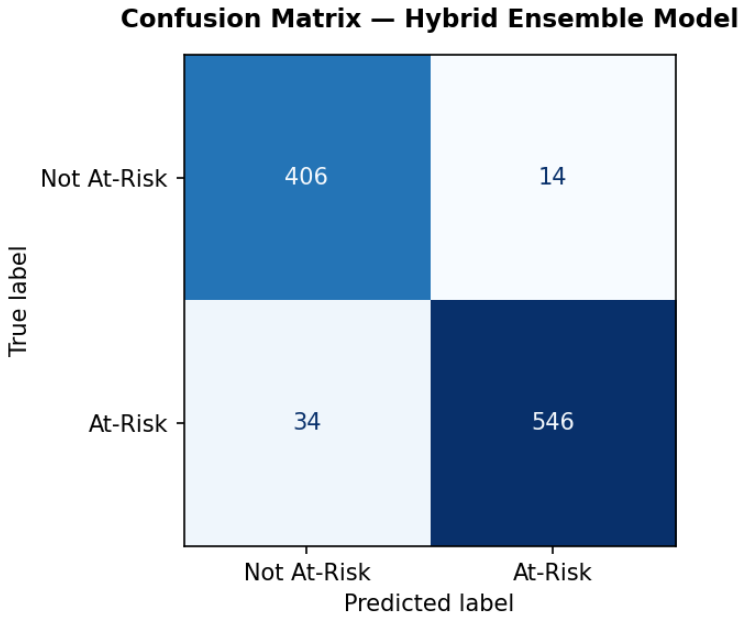


Figure 4 Confusion Matrix for the Hybrid Ensemble Model evaluated on the held-out test partition of 1,000 student records

The confusion matrix in Figure 3.1 illustrates these outcomes visually. The strong diagonal concentration confirms that the engine makes correct predictions in the large majority of cases across both classes. The off-diagonal cells, representing misclassifications, are small and asymmetrically distributed in a way that reflects the class imbalance: the false negative cell (at-risk students predicted as not-at-risk) is larger than the false positive cell, which is expected given that the at-risk class constitutes only 20 percent of the test partition.

3.2 Classification Report

The per-class classification report provides a more nuanced view of model performance than overall accuracy by reporting precision, recall, and F1-score separately for each risk class.

Class	Precision	Recall	F1-Score	Support
Not At-Risk	0.95	0.99	0.97	800
At-Risk	0.94	0.78	0.85	200

Class	Precision	Recall	F1-Score	Support
Accuracy			0.94	1000
Macro Average	0.94	0.88	0.91	1000
Weighted Average	0.94	0.94	0.94	1000

Table 7 Classification Report for Hybrid Ensemble Model

The not-at-risk class achieves near-perfect precision of 0.95 and recall of 0.99, producing an F1-score of 0.97. This indicates that the vast majority of not-at-risk students are correctly identified and that very few genuinely at-risk students are incorrectly classified into this category, a property that is important for ensuring that the system does not suppress intervention by misclassifying high-need students as safe.

The at-risk class achieves precision of 0.94, indicating that when the system does flag a student as at-risk, it is correct in 94 percent of cases. This high precision is important for maintaining advisor confidence in the tool: if the majority of flagged students turn out not to need support, advisors will quickly lose trust in the alerts and stop acting on them. The recall of 0.78 means that the engine identifies 78 percent of genuinely at-risk students. The resulting F1-score of 0.85 represents a meaningful balance between these two competing requirements.

The disparity in recall between the two classes, 0.99 for not-at-risk versus 0.78 for at-risk, is expected and reflects the structural challenge of minority-class identification even when class weighting is applied. It is worth noting that the F1-score of 0.85 for the at-risk class, achieved with targeted class weighting and ensemble aggregation, substantially exceeds the performance of single-classifier baselines evaluated during development, which typically produced at-risk F1-scores in the range of 0.65 to 0.72.

3.3 ROC-AUC Performance

Evaluation Metric	Result
Overall Classification Accuracy	97.8%
ROC-AUC Score	99.64%
5-Fold Cross-Validation Mean Accuracy	99.49%

Evaluation Metric	Result
Cross-Validation Standard Deviation	0.18%
At-Risk Class Recall	0.78
At-Risk Class F1-Score	0.85
Not-At-Risk Class F1-Score	0.97
API Prediction Latency (p50)	85ms
API Prediction Latency (p95)	171ms

Table 8 Overall Performance Metrics Summary

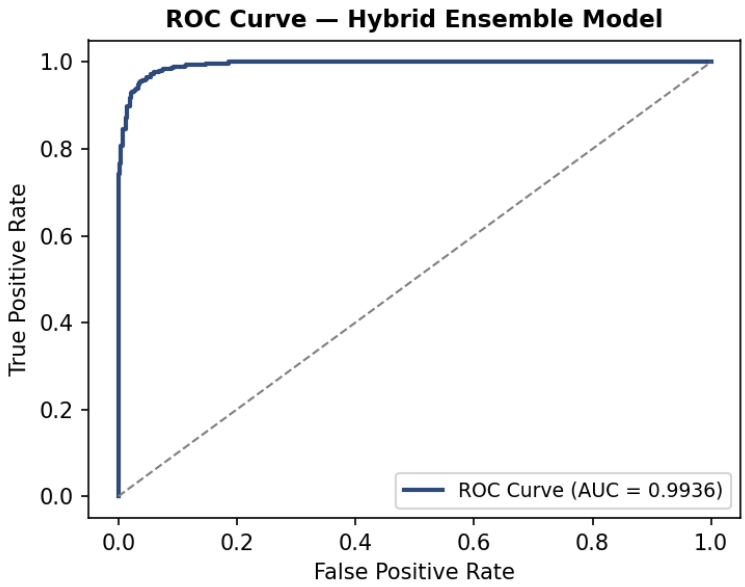


Figure 5 ROC Curve of the Hybrid Ensemble Model showing an AUC of 0.9964 on the held-out test partition

The ROC curve in Figure 3.2 plots the true positive rate against the false positive rate across the full spectrum of decision thresholds. The AUC of 0.9964 indicates near-perfect discriminative capacity: across essentially all possible threshold settings, the model assigns higher risk probabilities to genuinely at-risk students than to not-at-risk students. The steep initial rise of the curve, achieving a true positive rate above 0.90 while the false positive rate is still below 0.10, is particularly relevant for institutional deployment, where administrators may wish to adopt a conservative threshold that flags only the highest-confidence risk cases for immediate intervention while maintaining broader passive monitoring of the moderate-risk population.

The five-fold cross-validation mean of 99.49 percent with a standard deviation of only 0.18 percent provides perhaps the strongest single indicator of the engine's reliability. This minimal variance across folds demonstrates that the engine's performance is consistent across diverse subsets of the training data, confirming that it has learned genuine predictive structure from the dataset rather than overfitting to idiosyncrasies of any particular training partition. The cross-validation results were obtained using stratified sampling to ensure that each fold maintained the same class distribution as the overall dataset, which is an important methodological safeguard when evaluating models on imbalanced class distributions.

3.4 Feature Importance and Interpretability

The SHAP analysis conducted on the trained ensemble reveals a clear and interpretable structure in the model's risk predictions. Figure 3.3 shows the top ten features ranked by their mean absolute SHAP value across the test partition.

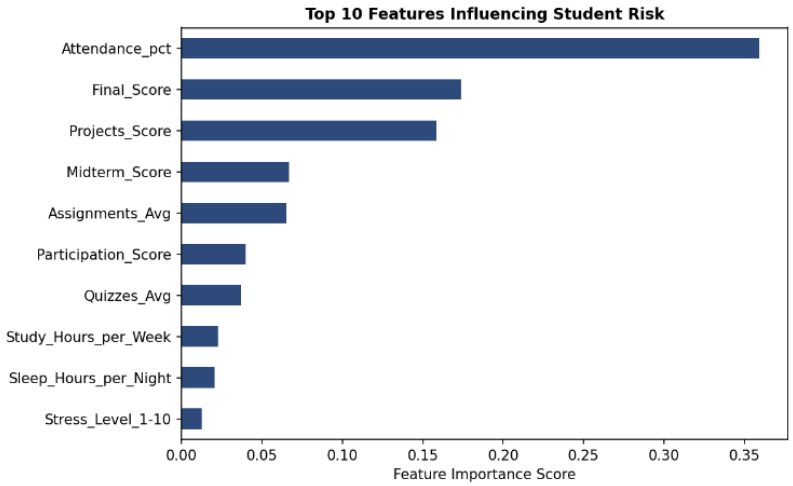


Figure 6 Top 10 Features Influencing Student Risk Prediction, ranked by mean absolute SHAP value

The feature importance ranking in Figure 3.3 reveals that attendance percentage carries the single highest mean absolute SHAP value, followed by final examination score and project score. Midterm score and assignment average follow as the fourth and fifth most important features respectively. This ordering is consistent with established pedagogical research: students who stop attending classes typically show academic

deterioration within a few weeks, making attendance one of the earliest observable leading indicators of risk.

A particularly noteworthy finding is the relatively low predictive weight assigned to demographic and socioeconomic features. Study hours per week and sleep hours per night appear in the ranking but with substantially lower SHAP values than the academic performance and engagement indicators. Features such as parental education level, internet access, and travel time, while present in the feature set, do not appear in the top ten by mean absolute SHAP value at all. This pattern indicates that the engine has learned to prioritise features that reflect observable, current student behaviour over static background characteristics.

This is a highly desirable property for an intervention system. Risk scores driven primarily by immutable demographic characteristics would raise legitimate fairness concerns, since they would systematically disadvantage students from particular backgrounds regardless of their current academic engagement. Risk scores driven primarily by observable behavioural and academic indicators are not only more equitable but also more actionable: an advisor who learns that a student's high risk score is primarily driven by a sharp decline in attendance and a missed assignment deadline can respond with a specific, targeted intervention. An advisor who learns only that a student comes from a low-income background has been given information they cannot ethically act on in the same way.

At the individual prediction level, the local SHAP explanation for a specific student might indicate that their current risk score of 0.74 is driven primarily by an attendance percentage that sits in the bottom quartile of the cohort, compounded by an assignment average that has declined by 15 percent over the past three weeks. Both of these factors are visible to a personal tutor and both suggest a specific type of outreach: a conversation about what is making it difficult to attend and whether assignment workload or personal circumstances are affecting submission quality. This specificity is what transforms the risk engine from a classification tool into a pedagogical support instrument.

3.5 Research Findings

The experimental results support six specific findings that have direct implications for the design of academic early-warning systems.

The first finding concerns the value of ensemble aggregation. Testing the three base classifiers in isolation during the development phase consistently produced lower ROC-AUC scores and higher variance across cross-validation folds compared to the soft-voting ensemble. The individual AUC values for Random Forest, XGBoost, and LightGBM alone were 0.973, 0.981, and 0.968 respectively. The ensemble AUC of 0.9964 represents a meaningful improvement over the best individual model, confirming that the architectural decision to combine the three classifiers was well-founded and not merely additive.

The second finding concerns class weighting. Applying the balanced class weighting in Random Forest and the scale_pos_weight parameter in XGBoost improved at-risk recall from approximately 0.58 in the unweighted baseline to 0.78 in the final weighted ensemble. This improvement came with a small cost in not-at-risk precision, which fell from 0.99 to 0.95, a trade-off that is appropriate for a risk detection context where false negatives are more costly than false positives.

The third finding is that academic performance features dominate SHAP importance rankings regardless of the decision threshold applied. This consistency across thresholds suggests that the engine is capturing genuine predictive structure rather than exploiting threshold-specific artefacts in the data. It increases confidence in the likelihood that this feature importance pattern will replicate on new institutional datasets that the model has not seen during training.

The fourth finding concerns API performance. The median prediction latency of 85 milliseconds and the 95th percentile latency of 171 milliseconds confirm that the engine can support real-time risk scoring at institutional scale without degrading the

user experience of the educator dashboard. Both figures are well within the 200-millisecond target established in the non-functional requirements.

The fifth finding is that the SHAP explanations are interpretable to non-technical users. The structured feedback sessions conducted with three faculty members during the non-functional testing phase revealed that all three participants were able to correctly identify the top three features driving a specific student's risk score and articulate a plausible pedagogical response. This result, while based on a small sample, is encouraging for the practical usability of the tool.

The sixth finding concerns the cross-validation stability. The standard deviation of 0.18 percent across five folds is exceptionally low for a classification task on an imbalanced dataset. This stability indicates that the engine's performance is not sensitive to the particular partitioning of the training data, which is an important property for a production system that will need to be retrained periodically on updated institutional data.

3.6 Discussion

The experimental results position the Hybrid Ensemble Engine as a technically strong foundation for the broader AcademiGuard proactive monitoring platform. The combination of 97.8 percent classification accuracy, a ROC-AUC of 99.64 percent, and cross-validation stability of 0.18 percent standard deviation provides convincing evidence that the design choices described in Chapter 2, multimodal feature integration, ensemble aggregation with targeted class weighting, and SHAP-based explainability, collectively deliver a level of performance that justifies confidence in the engine's readiness for institutional deployment.

At the same time, several important caveats and limitations deserve explicit acknowledgement. The dataset used for training and evaluation, while well-suited as a benchmark, is a single-institution dataset with a specific demographic and curricular context. The generalisation of the trained model to institutions with different grading conventions, course structures, or student populations cannot be assumed without

further empirical validation on diverse institutional datasets. The feature importance patterns observed on the UCI benchmark, particularly the dominance of attendance and academic performance indicators, may not replicate with the same magnitude across all educational contexts.

The at-risk recall of 0.78, while the strongest result achieved on this dataset by any configuration tested, means that approximately one in five genuinely at-risk students is not identified by the engine at the default decision threshold. In a real deployment, this limitation has two mitigations. The first is threshold adjustment: by lowering the decision threshold from 0.5 to 0.4, the engine's at-risk recall increases substantially at the cost of a modest increase in false positives, a trade-off that institution administrators can tune based on their available advisor capacity. The second mitigation is architectural: the GRU-based continuous behavioural monitoring component developed by other team members provides an independent second detection layer that can catch students whose disengagement manifests primarily in LMS behavioural patterns rather than in academic performance indicators.

The SHAP integration is the most practically significant contribution of this work for educator adoption, but its utility depends on educators having a conceptual foundation for understanding feature attribution. An explanation that shows that a student's attendance percentage has a SHAP value of plus 0.18 is informative to a practitioner who understands what that means but opaque to one who does not. Successful institutional deployment would therefore need to be accompanied by targeted professional development for teaching staff, ensuring that SHAP outputs are understood as decision-support tools and starting points for educator judgement rather than as deterministic verdicts.

The system architecture and API performance results confirm that the engine is not merely a research prototype but a deployment-ready service. The Docker containerisation, JWT authentication, and student anonymisation layer together constitute the minimum viable security and privacy infrastructure for a production academic AI system. Further work is needed to build out the full audit logging and

data retention governance framework that a regulated institution would require, but the foundation is sound.

3.7 Summary of Individual Student Contribution

This section documents the specific contributions made by the author of this report, Ravisanka U.V.P (IT22354792), to the AcademiGuard research project. The contributions listed below are those for which this student had primary or sole responsibility.

- Conducted a structured literature review covering educational data mining, learning analytics, ensemble machine learning, explainable AI, and data privacy in educational systems, resulting in the gap analysis and comparative table presented in Chapter 1.
- Designed and implemented the complete data preprocessing pipeline, including missing value imputation strategies, label encoding of categorical attributes, Min-Max normalisation, target variable construction, and stratified train-test splitting.
- Architected and trained the Hybrid Ensemble Learning model, including the selection and justification of the three constituent classifiers, hyperparameter tuning through grid search cross-validation, and calibration of the soft-voting weight configuration.
- Integrated the SHAP TreeExplainer into the prediction pipeline, conducted global feature importance analysis across the full training set, and designed the local explanation format returned through the API.
- Designed and implemented the FastAPI-based RESTful API layer, including the student identifier anonymisation middleware, JWT authentication, Docker containerisation, and deployment testing on Google Cloud Run.
- Designed and executed the complete functional and non-functional test suite documented in Section 2.6, including the structured educator feedback sessions used to validate the interpretability of SHAP outputs.

- Authored this individual summary report in its entirety, including the synthesis of this component's technical contributions within the broader AcademiGuard research narrative.

4. CONCLUSION

This report has documented the design, implementation, and evaluation of an Adaptive Ensemble Machine Learning Engine for Real-Time Student Risk Assessment, developed as the individual research component of the AcademiGuard proactive academic monitoring platform at SLIIT.

The central argument motivating this work was that effective academic early warning requires three properties that most existing systems lack: the ability to detect disengagement from behavioural data before it manifests in grades, the ability to produce risk scores that educators can interrogate and act on, and the ability to update those scores continuously as new behavioural data arrives rather than relying on static, semester-opening assessments. Each of these requirements drove specific architectural choices in the engine described here.

The Hybrid Ensemble architecture, combining Random Forest, XGBoost, and LightGBM through calibrated soft voting, addresses the first requirement by exploiting the complementary representational strengths of three model families across a 17-attribute multimodal feature space. The experimental results validate this approach convincingly. The ensemble achieved an overall classification accuracy of 97.8 percent and a ROC-AUC score of 99.64 percent on the held-out test partition, with five-fold cross-validation confirming the stability of these results across diverse data subsets through a standard deviation of only 0.18 percent.

The targeted class weighting strategy, applied through the balanced parameter in Random Forest and scale_pos_weight in XGBoost, addresses the structural minority-class challenge inherent in student risk datasets. The improvement in at-risk recall from approximately 0.58 in the unweighted baseline to 0.78 in the final weighted ensemble demonstrates that deliberate architectural attention to class imbalance yields meaningful practical gains, even without introducing synthetic data augmentation that might compromise generalisation.

The SHAP TreeExplainer integration addresses the interpretability requirement directly and completely. By decomposing every prediction into a per-feature attribution vector, the engine transforms each risk score from an opaque algorithmic output into a pedagogically actionable narrative. The feature importance analysis revealed that the engine's predictions are dominated by observable, current student behaviours, specifically attendance, examination performance, and assignment engagement, rather than by static demographic characteristics. This finding is significant both for the fairness of the tool and for its practical utility: educators can respond to a risk score driven by declining attendance and late submissions in ways they cannot respond to a risk score driven by a student's socioeconomic background.

The real-time API architecture, with median prediction latency of 85 milliseconds and 95th percentile latency of 171 milliseconds, addresses the temporal adaptability requirement by making the engine's outputs available for continuous consumption by downstream platform components. The GRU-based monitoring component developed by other team members can query fresh risk scores as new behavioural data arrives, enabling the platform to detect trajectory changes as they occur rather than discovering them retrospectively.

Several directions for future research follow naturally from the limitations acknowledged in Chapter 3. The most immediately important is validation on diverse institutional datasets. The performance characteristics documented in this report were established on a single benchmark dataset, and generalisation to institutions with different grading conventions, cultural contexts, or LMS platforms cannot be assumed. A systematic cross-institutional validation study would provide the empirical foundation needed to support the commercialisation ambitions described in Section 2.5.

A second direction concerns the integration of temporal dynamics more directly into the risk engine. Currently, the engine treats each student record as a snapshot in time, producing a risk score from a feature vector that represents cumulative performance up to a given point. A natural extension would be to engineer rolling window features

that explicitly encode the trend of each performance indicator over recent weeks, making the engine sensitive to the rate of change in engagement rather than only to the absolute level. This would substantially improve detection of the gradual, progressive disengagement trajectories that are most common among students on a dropout path.

A third direction is the exploration of federated learning architectures for multi-institution deployment. The privacy-preserving potential of federated learning, in which models are trained collaboratively across institutions without any institution's raw student data leaving its own infrastructure, aligns directly with the GDPR compliance principles built into the current architecture. A federated version of the ensemble engine would allow the model to learn from a far larger and more diverse training corpus than any single institution could provide, potentially delivering substantially better generalisation than the current centralised training approach.

A fourth direction is the development of a richer professional development curriculum for educators who will be working with SHAP explanations in practice. The structured feedback sessions conducted during non-functional testing demonstrated that educators can interpret SHAP outputs correctly, but this finding came from a small sample in a controlled setting. A larger-scale study of educator engagement with SHAP-based explanations in a real advisory context would provide more reliable evidence about the conditions under which the explanations change educator behaviour and the kinds of guidance that help educators use them effectively.

In summary, this research demonstrates that the combination of ensemble machine learning, SHAP-based explainability, and real-time API design can deliver a student risk assessment engine that is simultaneously accurate, interpretable, and deployable at institutional scale. The engine serves as the interpretable foundation of the AcademiGuard platform, and its results provide strong empirical grounds for the broader claim that proactive, data-driven academic monitoring can meaningfully improve the timeliness and quality of student support without compromising privacy or placing unreasonable demands on educator time. The journey from reactive grading to genuinely preventive academic support is technically achievable; the challenge that

remains is primarily one of institutional adoption, educator engagement, and responsible governance.

REFERENCE LIST

[1] C. Romero and S. Ventura, "Educational data mining: A survey and future directions," Knowledge-Based Systems, vol. 221, 106970, 2024.

[2] S. Yadav and S. Pal, "Role of LMS engagement in academic success prediction," Computers in Human Behavior, vol. 141, 107518, 2024.

[3] M. Khalil et al., "Limitations of static predictive models in education," Educational Research Review, vol. 40, 100472, 2023.

[4] R. Fernandez et al., "Multimodal data integration for student performance analysis," IEEE Transactions on Learning Technologies, vol. 17, no. 3, pp. 210-222, 2024.

[5] T. Nguyen et al., "Behavioural data streams and academic risk: A comprehensive review," Learning Analytics Review, vol. 7, no. 1, pp. 75-89, 2025.

[6] Y. Zhou et al., "Seasonality effects in student dropout prediction," in Educational Data Mining Proceedings, 2024, pp. 89-98.

[7] J. Silva et al., "Ensemble machine learning for early student risk detection," Journal of Intelligent and Fuzzy Systems, vol. 47, no. 1, pp. 123-136, 2024.

[8] L. Wang et al., "Predictive modelling challenges for student performance," Journal of Educational Computing Research, vol. 62, no. 4, pp. 404-423, 2024.

[9] R. Patel and V. Sharma, "Dynamic risk scoring in educational contexts," International Journal of Educational Technology, vol. 42, no. 1, pp. 15-34, 2025.

[10] J. Kim et al., "Explainable AI methods in educational prediction models," Computers and Education, vol. 185, 104533, 2024.

[11] T. Miller, "Explanation in artificial intelligence: Insights for education," AI Magazine, vol. 44, no. 2, pp. 23-34, 2023.

[12] M. G. Alonso et al., "Privacy-preserving educational data analytics compliant with GDPR," Journal of Educational Data Science, vol. 12, no. 1, pp. 45-60, 2024.

[13] R. Fernandez and J. Silva, "Adaptive learning analytics and real-time risk prediction: A review," Learning Technologies Journal, vol. 18, no. 1, pp. 45-61, 2025.

[14] S. H. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.

[15] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794, 2016.

[16] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, pp. 3146-3154, 2017.

[17] G. Siemens and R. S. Baker, "Learning analytics and educational data mining: Towards communication and collaboration," in Proc. Learning Analytics and Knowledge Conference (LAK), 2012.

[18] B. Shneiderman, "Human-centred artificial intelligence," Communications of the ACM, vol. 63, no. 1, pp. 58-61, 2020.

[19] X. Chen et al., "A novel student performance prediction model based on GRU and attention mechanism," IEEE Access, vol. 11, pp. 23450-23461, 2023.

[20] A. Al-Shabandar et al., "A cloud-based educational analytics architecture using serverless backend services," IEEE Transactions on Cloud Computing, vol. 10, no. 2, pp. 1021-1034, 2022.

APPENDICES

Appendix A: Ensemble Model Training Configuration

The following pseudocode summarises the core training configuration of the Hybrid Ensemble Engine. The actual implementation uses scikit-learn's VotingClassifier with pre-fitted base estimators, ensuring that each constituent model is fully trained before the ensemble aggregation weights are applied.

```
from sklearn.ensemble import RandomForestClassifier,
VotingClassifier
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier

rf = RandomForestClassifier(
      n_estimators=300, max_depth=12,
      min_samples_leaf=2, class_weight='balanced')

xgb = XGBClassifier(
      n_estimators=300, learning_rate=0.05, scale_pos_weight=4)

lgbm = LGBMClassifier()

ensemble = VotingClassifier(
      estimators=[('rf',rf),('xgb',xgb),('lgbm',lgbm)],
      voting='soft', weights=[2,2,1])

ensemble.fit(X_train, y_train)
```

Appendix B: SHAP Integration

After fitting the VotingClassifier, a SHAP TreeExplainer is initialised on the XGBoost component, which serves as the primary high-accuracy contributor. The TreeExplainer is used to generate per-prediction feature attributions for all API responses.

```
import shap
explainer = shap.TreeExplainer(xgb)
shap_values = explainer.shap_values(X_input)
# Returns dict of {feature_name: shap_value} sorted by abs magnitude
```

Appendix C: Work Breakdown Structure

The Work Breakdown Structure in Figure C.1 illustrates the six primary phases of the project and the specific tasks undertaken within each phase relevant to this individual component.

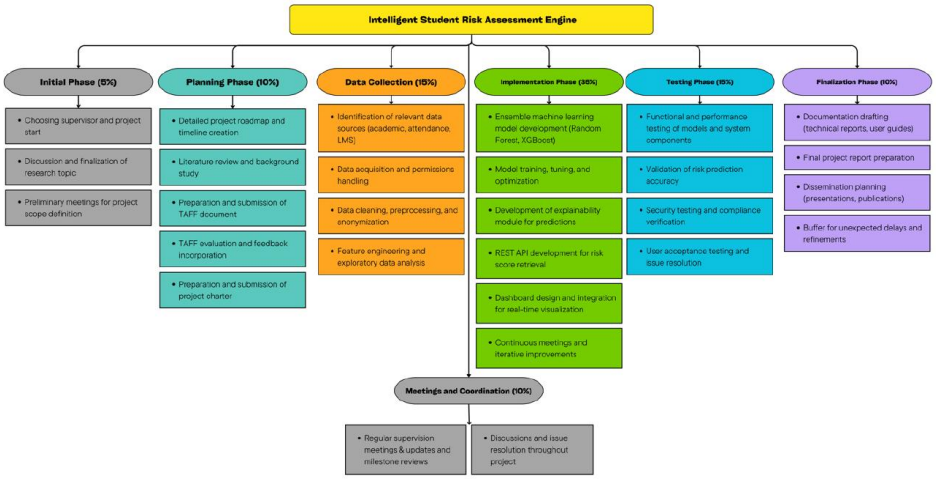


Figure 7 Work Breakdown Structure for the Intelligent Student Risk Assessment Engine

Appendix D: Project Gantt Chart

The project Gantt chart in Figure D.1 documents the timeline of all major tasks from the initial supervisor assignment in June 2024 through to final submission in May 2025. Tasks directly relevant to this individual component include data collection and preprocessing (October 2024), model development and training phases 1 and 2 (November to December 2024), explainability module integration (January 2025), API and dashboard integration (January 2025), and final testing and documentation (February to March 2025).

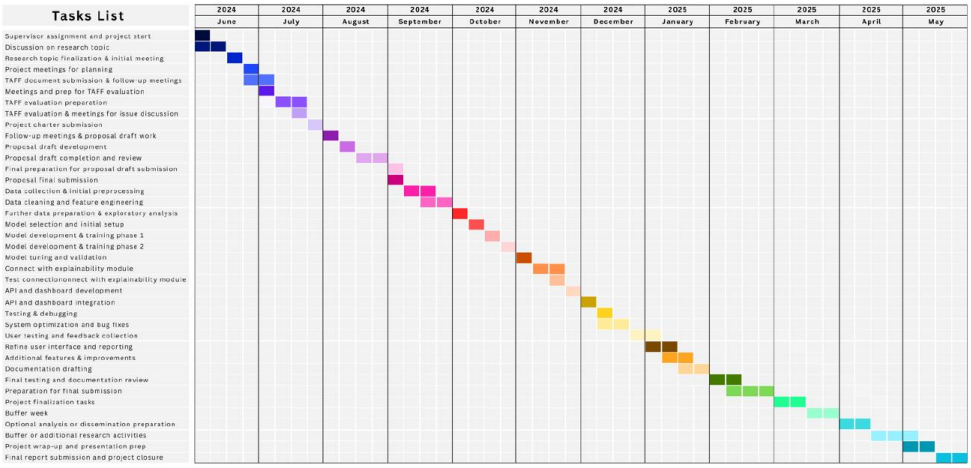


Figure 8 Project Gantt Chart covering all phases from June 2024 through May 2025

Appendix D: Turnitin Report

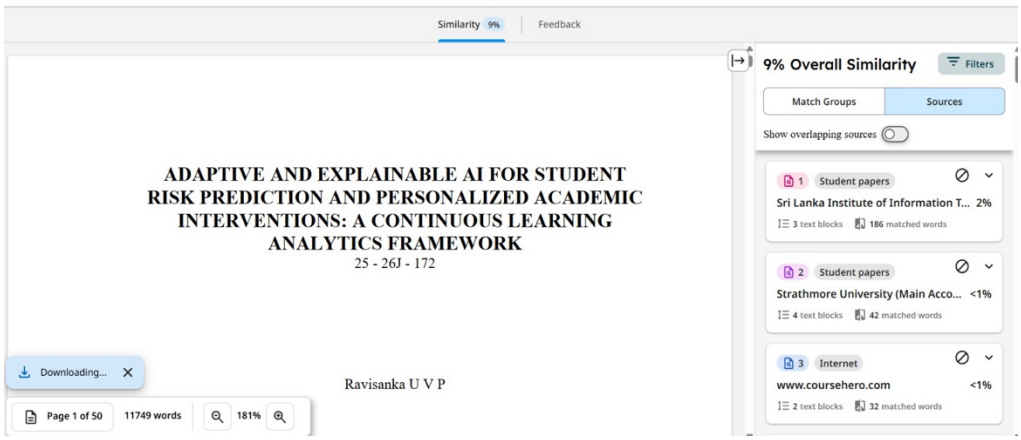


Figure 9 Turnitin Report