International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

ARMS: An Attendance-Linked Early Risk Prediction and Academic Record Management Framework for Higher Education

Dr. S.R. Khare
Department of Information Technology
Priyadarshini College of Engineering
Nagpur , Maharashtra , India

Mrs. Varsha Katiwal
Department of Information Technology
Priyadarshini College of Engineering
Nagpur , Maharashtra , India

Divyansh Hedau
Department of Information Technology
Priyadarshini College of Engineering
Nagpur , Maharashtra , India

Kartik Singh
Department of Information Technology
Priyadarshini College of Engineering
Nagpur , Maharashtra , India

Piyush Pandey
Department of Information Technology
Priyadarshini College of Engineering
Nagpur , Maharashtra , India

Vaibhav Singh
Department of Information Technology
Priyadarshini College of Engineering
Nagpur , Maharashtra , India

Abstract—Academic institutions commonly compute results in spreadsheets, so struggling students surface only after failures appear in the gradebook. This paper presents ARMS (Academic Record and Management System), a web-based framework that automates result computation, consolidates Semester 1–8 records, and estimates student risk from academic performance and attendance together. Its central component is an interpretable score $R = 0.55(100 - P) + 0.45(100 - A)$, where P is mean percentage and A is mean attendance. A policy override raises the risk whenever attendance drops below the 75% exam-eligibility limit typical of these institutions. In a seeded ten-student case study, a fail-count baseline missed four students whose risk stemmed from low attendance or declining marks, and ARMS flagged all four. ARMS did under-flag two students with repeated failures. On a simulated cohort of 2,000 students under five-fold cross-validation, the ARMS score reached an AUC of 0.907. This is close to logistic regression (0.915) and higher than random forest (0.860) and a fail-count rule (0.727). An ablation shows that attendance adds little ranking power beyond performance in this simulation (AUC 0.908 versus 0.916 at the best weighting). Its value therefore lies in early availability and policy compliance rather than discrimination. The default threshold favours recall (0.93) over precision (0.27), and sweeping the threshold places the F1 peak near 45. The system scores 10,000 students in about 0.26 s. All results come from simulation and do not establish real-world accuracy.

Index Terms—early warning system, learning analytics, attendance, student risk prediction, result management, Flask.

INTRODUCTION

Two problems recur in college-level academic administration. The first is result processing, which is often manual or spreadsheet-based and therefore slow and error-prone. The second is late intervention: a student is usually identified as at risk only after accumulating failures, when recovery is difficult. Early-warning practice treats course grades and attendance as the most common indicators [1], and recent work finds that historical performance, attendance, and assignment behaviour are strong predictors of failure [2]. Yet many practical result systems record these values passively and never combine them.

In Indian engineering colleges the problem also has a regulatory edge. A student with strong marks can still be barred from an examination for falling below a minimum attendance, typically 75%, and a purely grade-based alert will miss this. A useful tool must therefore treat attendance as a first-class signal. It must also stay understandable to teachers who are not data scientists, and it must run on modest infrastructure.

This paper makes the following contributions:

- a unified result-management pipeline that validates imported data, applies weighted grading, computes GPA and CGPA, ranks students, and produces PDF report cards and consolidated Semester 1–8 transcripts;
- an open implementation of about 1,800 lines of Python whose JSON store exposes a MongoDB-style API, so the system deploys with no external infrastructure; and
- an evaluation that combines boundary tests, a case study, a simulation with baselines, ablation, sensitivity and robustness analyses, and runtime measurements, and states its limitations openly.

Section II reviews related work. Section III describes the architecture, and Section IV covers the methods. The implementation appears in Section V and the evaluation in Section VI. Section VII discusses limitations, and Section VIII concludes the paper.

II. RELATED WORK

Educational data mining. Reviews of educational data mining and academic-success prediction cover the techniques, data sources, and recurring pitfalls of the field [3], [4]. Two lessons recur. Prior performance is a dependable predictor, and a model is useful only if teachers can act on its output.

Early warning systems. Course Signals at Purdue is an influential example of learning analytics applied to student success [5]. One system for online higher education couples predictive models with dashboards and lets teachers send early feedback [6]. A Taiwanese study combined historical performance, attendance, and assignment submission to reach a balanced AUC of 0.85, and personalised support helped 73% of the flagged students pass [2]. These systems typically depend on learning-platform logs, which a conventional engineering college may not have.

Simple rules versus machine learning. A district-level evaluation found that a prior-performance early-warning flag and a machine-learning risk score were similarly accurate at the same flag rate; the simple flag was cheaper to deploy but less informative about who is at greatest risk [7]. This motivates ARMS: where labelled outcomes are scarce, a transparent weighted score is a reasonable starting point, provided it is calibrated.

Gap. Result-management software typically stores marks and produces reports, whereas early-warning models are usually separate analytical tools. ARMS embeds a transparent risk indicator, including an explicit attendance-eligibility rule, inside the gradebook workflow that teachers already use.

III. SYSTEM ARCHITECTURE

ARMS is a Flask [8] application organised as blueprints over a shared calculation engine and data layer (Fig. 1). Table I lists the modules and their sizes.

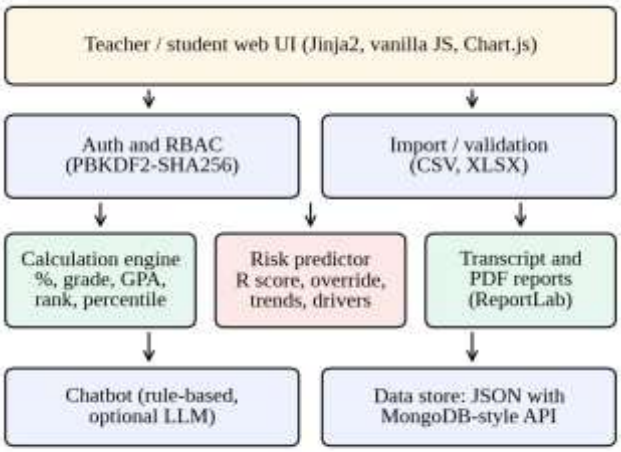


Fig. 1. ARMS architecture. The risk predictor and calculation engine share the same stored records.

TABLE I. MODULES (PYTHON LINES OF CODE)

Module	Responsibility	LOC
calc_engine	Totals, %, grade, GPA, rank	124
validators	ID, mark and attendance checks	62
student_predictor	Risk score, level, trends	167
transcript	Semester 1–8 consolidation	74
pdf_exporter	Report cards, transcripts	211
import_export	CSV/XLSX import, export	91
chatbot_handler	Dual-persona assistant	142
auth	Login, sign-up, RBAC	181
db	JSON store, Mongo-style API	127
Other (app, config, …)	Routing, insights, seed data	625

Student records are stored as documents keyed by college ID, each holding results nested by session and exam. Every exam row carries subject scores, percentage, grade, GPA, and attendance. This nesting lets the transcript and risk modules read the same data the report generator writes. Role-based access separates teachers from students, and passwords are hashed with PBKDF2-SHA256. Teacher sign-up validates the ID format, uniqueness, and password strength. A dual-persona chatbot answers questions offline with rule-based logic and can switch to an external language model when an API key is configured.

The teacher dashboard has a spreadsheet-style editor whose at-risk panel updates on every save. It also includes a report generator with chart options and bulk PDF export, and an academic-record search. The student portal shows results, attendance, and a risk badge with its drivers, along with a GPA-trend chart and a downloadable transcript.

IV. METHODOLOGY

A. Grading Engine

For subject j with score x_j, maximum m_j, and weight w_j, the overall percentage is

$$P = 100 \cdot \Sigma\, w_j\, x_j \,/\, \Sigma\, w_j\, m_j \qquad (1)$$

Letter grades and grade points follow a threshold lookup on percentage. GPA is the weight-averaged grade point, and CGPA is the mean GPA across sessions. Rank and percentile are computed within the class by sorting on percentage.

B. Data Validation

Before results are finalised, the validator checks the imported sheet for missing or duplicate student IDs, IDs outside an allowed pattern, missing names, missing or non-numeric marks, negative marks, marks above the subject maximum, and attendance outside 0-100. It returns each issue as a readable row-level message, so a teacher can fix the sheet without debugging a stack trace.

C. Attendance-Linked Risk Score

Let Pavg be a student's mean percentage over all recorded exams, and let Aavg be the mean attendance (taken as 100% if never recorded). The risk score is

$$R = w_p(100 - P_{avg}) + w_a(100 - A_{avg}) \qquad (2)$$

with wp = 0.55 and wa = 0.45 in the deployed system. Levels are High for $R \geq 60$, Medium for $35 \leq R < 60$, and Low otherwise. These weights and thresholds are design choices, and Sections VI-E and VI-F examine their sensitivity.

D. Policy Override

Many institutions require about 75% attendance for exam eligibility, and a weighted average can mask a breach. ARMS therefore applies this rule: if Aavg < 65, the level is High; otherwise, if Aavg < 75 and the level is Low, it becomes Medium. A student is at risk when the final level is Medium or High. Algorithm 1 gives the complete procedure.

Algorithm 1. Risk assessment for one student

```
input: results[session][exam] rows with pct, attendance, grades
P <- mean(pct over all exams)
A <- mean(attendance) if recorded else 100
R <- 0.55*(100-P) + 0.45*(100-A)
level <- High if R>=60; Medium if R>=35; else Low
if A < 65: level <- High
else if A < 75 and level = Low: level <- Medium
trend <- compare last vs first value (+/-5 points)
drivers <- {A<75, fail count>0, P<40, falling attendance}
return (R, level, at_risk = level in {Medium, High}, drivers)
```

The procedure visits each stored exam row once, so its cost is linear in the number of rows for a student and linear in class size for a class-wide scan.

E. Trends and Explainability

Performance and attendance trends are labelled declining, improving, or stable by comparing the first and last recorded values against a ±5-point margin. The system also emits risk drivers, such as low attendance, number of subject failures, low average, or declining attendance, which students see in their portal and which the chatbot cites when teachers ask who is at risk.

F. Transcript Consolidation

Results are stored per session and exam. The transcript module extracts the numeric part of each session label, so Semester 1–8 sort numerically rather than alphabetically, and then computes per-semester GPA and running CGPA. One function serves both the teacher and student views, so the two always agree.

V. IMPLEMENTATION AND DEPLOYMENT

The backend is Python with Flask blueprints. The frontend uses vanilla JavaScript, Jinja2 templates, and Chart.js, and PDF output is generated with ReportLab. Spreadsheet import and export use pandas and openpyxl. The JSON store implements find, find_one, insert_one, and update_one with upsert, so the whole application runs without an external service, and a configuration flag switches the same calls to MongoDB.

The application is served by Gunicorn on a free hosting tier. That tier has an ephemeral filesystem, so stored data are lost on restart. For demonstrations we therefore run an idempotent seeding script at start-up that upserts ten students with eight semesters each. Production use would need MongoDB or a persistent disk. The interface supports light and dark themes and a floating assistant on both teacher and student sides.

Deployment experience. Two practical issues are worth recording for others deploying small Flask systems. First, if the repository root already contains the application, a hosting service configured with a non-existent root subdirectory fails at the clone step; clearing that setting resolves it. Second, the free tier gives no interactive shell access, so one-off seeding commands cannot be run by hand. Chaining the seeding script before the web server in the start command removes the need for a shell and keeps the demonstration data across restarts.

Typical workflow. A teacher creates a sheet from a blank or template layout, or uploads a spreadsheet. The validator reports problems, and the engine then computes grades and ranks. As marks and attendance are typed, the at-risk panel highlights students. Publishing writes the rows to each student's record, so the portal, transcript, and chatbot all read the same data.

VI. EXPERIMENTAL EVALUATION

A. Scope and Protocol

No real student outcome data are used. We report boundary tests of the deployed predictor,

We report boundary tests of the deployed predictor, a case study on ten seeded students, and a simulation study with baselines. None of these measures real-world accuracy. All code and seeds are in the project repository (random seed 42). The simulation models come from scikit-learn [9], and the random forest [10] uses 200 trees. AUC is the area under the ROC curve [11].

B. Boundary Verification

We called the deployed predictor with inputs on either side of each rule. Table II confirms that the override acts exactly at 65% and 75% attendance, and that the score threshold of 35 corresponds to an average of about 36% with full attendance.

TABLE II. BOUNDARY TESTS OF THE DEPLOYED PREDICTOR

Avg %	Att. %	R	Level	Rule
90	64.9	21.3	High	< 65
90	65.0	21.2	Medium	< 75
90	74.9	16.8	Medium	< 75
90	75.0	16.8	Low	none
40	100	33.0	Low	none
34	100	36.3	Medium	R ≥ 35

C. Case Study on a Seeded Cohort

Ten synthetic students, each with eight semesters and three subjects, follow deliberately designed profiles. The baseline is the original rule, which flags a student with two or more failed subjects. Table III and Fig. 2 show the outcome.

TABLE III. CASE-STUDY RESULTS (SYNTHETIC COHORT)

ID	Avg %	Att. %	Fails	R	ARMS	Baseline
S2001	82.8	92.5	0	12.8	Low	No
S2002	33.0	55.5	12	56.9	High	Yes
S2003	62.3	69.5	0	34.5	Medium	No
S2004	63.5	78.0	0	30.0	Low	No
S2005	78.0	49.4	0	34.9	High	No
S2006	61.2	80.8	0	30.0	Low	No
S2007	65.1	72.4	0	31.6	Medium	No
S2008	61.6	77.5	5	31.2	Low	Yes
S2009	75.5	84.4	3	20.5	Low	Yes
S2010	54.1	76.0	1	36.1	Medium	No

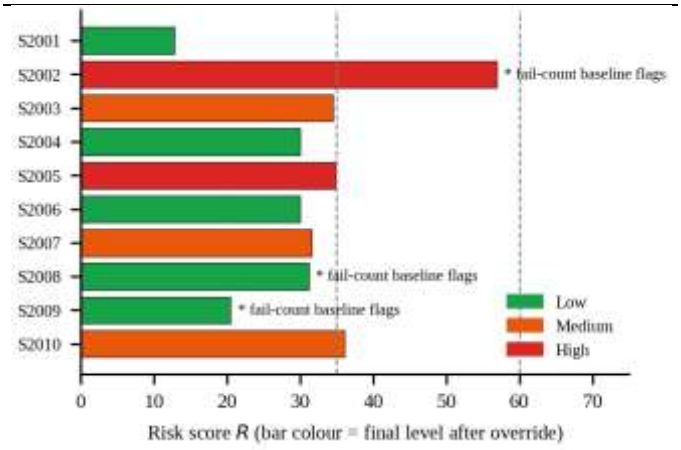


Fig. 2. Case-study risk scores. Dashed lines mark the 35 and 60 thresholds.

The fail-count baseline fails to capture S2003, S2005, S2007, and S2010, as their vulnerability stems from poor attendance or downward academic trajectories rather than recorded backlogs. ARMS flags all four. For S2005 (78.0% average, 49.4% attendance), the escalation to High risk occurs exclusively via the regulatory override; the base score of R = 34.9 sits just beneath the Medium cutoff. By contrast, S2008 (5 failures) and S2009 (3 failures) remain categorized as Low risk because the formulation of R omits historical course failures. A composite union strategy—flagging an individual whenever either detector triggers—would identify seven of the ten students, leaving only S2001, S2004, and S2006 unflagged. We note this hybrid formulation as an operational design recommendation rather than an empirically validated result.

D. Simulation Study

Setup. To evaluate classification performance, we synthesized a cohort of 2,000 students modeled on latent academic aptitude and behavioral engagement. These

underlying traits govern three pre-examination indicators: mean internal percentage, cumulative attendance, and number of course failures. The target label—end-semester failure, defined as a final score under 40%—derives from these latent parameters combined with stochastic noise, yielding a failure prevalence of 11.8%.

TABLE IV. SIMULATION RESULTS (5-FOLD CV MEAN)

Method	Prec.	Recall	F1	AUC
Fail-count rule (≥ 2)	0.688	0.275	0.392	0.727
ARMS, no override	0.271	0.928	0.419	0.907
ARMS + override	0.227	0.949	0.366	0.907
ARMS calibrated	0.247	0.899	0.388	0.915
ARMS + fail union	0.226	0.949	0.366	0.908
Logistic regression	0.657	0.411	0.505	0.915
Random forest	0.624	0.466	0.531	0.860

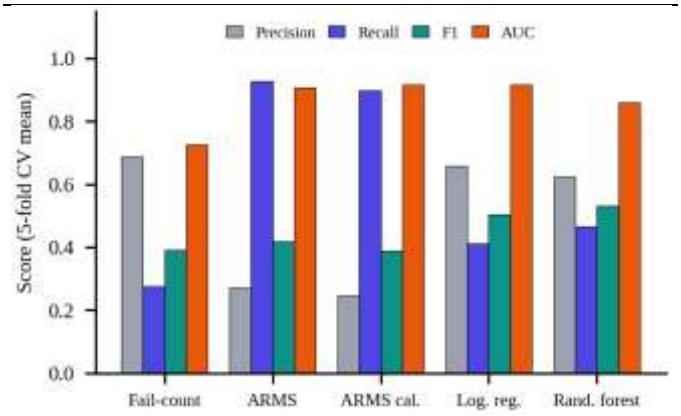


Fig. 3. Simulation results (five-fold CV mean).

The transparent heuristic ranks student vulnerability comparably to logistic regression (AUC 0.907 vs. 0.915) while outperforming both the random forest classifier (0.860) and the rudimentary fail-count rule (0.727). This empirical parity aligns directly with prior evidence demonstrating that lightweight, rule-based indicators can match parameterized machine learning models without incurring deployment overhead [7].

The predetermined cutoff of 35 prioritizes recall, successfully capturing 93% of failing students at the expense of a high false-positive rate (precision 0.27). Across all five folds, cross-validated grid search converged uniformly on $w_p = 0.70$ paired with an operating cutoff of 45. Enforcing the attendance override depresses precision further—from 0.271 to 0.227—because our ground-truth label reflects terminal academic failure rather than regulatory disqualification. Consequently, the override functions primarily as a compliance safeguard rather than a statistical predictor.

E. Ablation and Weight Sensitivity

Table V benchmarks isolated individual predictors against the composite formulation across the entire simulated cohort, while Fig. 4 traces the trajectory of the classification AUC as the performance weight w_p spans from 0 to 1.

TABLE V. ABLATION (AUC, FULL SIMULATED COHORT)

Score	AUC
Performance only (100 − P)	0.908
Attendance only (100 − A)	0.782
Failed-subject count only	0.727
ARMS, $w_p = 0.55$ (deployed)	0.907
ARMS, $w_p = 0.70$	0.916

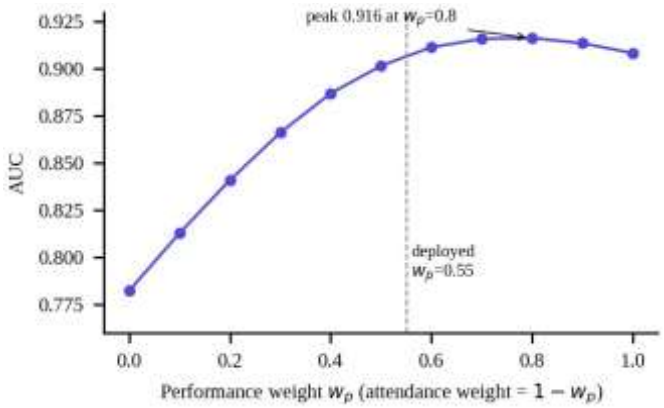


Fig. 4. AUC as the performance weight varies. The curve is flat between 0.6 and 1.0.

This finding challenges our initial design premise: performance alone matches the deployed score's ranking power in this simulation, with optimal weights (w_p = 0.7–0.8) gaining only ~0.008 AUC. Attendance alone predicts less accurately (0.782) but remains far better than failure counts (0.727). Rather than yielding large ranking gains here, attendance proves valuable because it is recorded weeks before marks , enforces regulatory eligibility , and catches disengaged students with healthy grades. Validating its additive value on real data remains on empirical questions.

F. Threshold Sensitivity

Fig. 5 illustrates the sensitivity of the deployed 0.55/0.45 score across decision thresholds. Elevating the cutoff from 35 to 40 and 45 drops recall from 0.93 to 0.80 and 0.56, while precision climbs from 0.27 to 0.42 and 0.57. F_1 reaches an optimal 0.56 at threshold 45, remaining close at 0.55 for 40. Institutions prioritizing detection over false alarms should maintain the cutoff near 35; those constrained by advising capacity should adjust it toward 40–45.

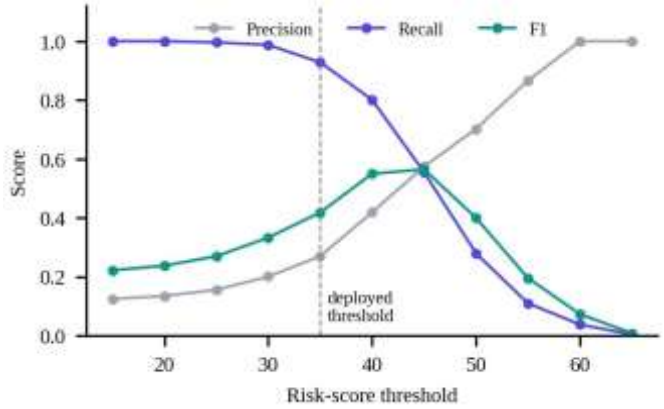


Fig. 5. Precision, recall and F1 as the risk threshold varies (no override).

G. Robustness to Label Noise

We repeated the cross-validated comparison under progressively higher noise in the final-examination outcome. As detailed in Table VI, all models degrade as label noise grows, yet ARMS tracks within 0.01 AUC of logistic regression across every noise level, whereas the random forest model remains consistently the lowest.

TABLE VI. CV AUC VERSUS OUTCOME NOISE

Noise σ	Prev.	ARMS	LogReg	RF
5	0.092	0.955	0.960	0.940
9	0.118	0.907	0.915	0.860
14	0.161	0.840	0.847	0.788
20	0.212	0.777	0.783	0.720

H. Runtime

We benchmarked execution runtimes for the grading engine and risk predictor across synthetic cohorts evaluated across five subjects. Table VII lists single-run wall-clock latency measured in the development sandbox; absolute execution speeds will naturally vary with underlying hardware. Computational overhead scales approximately linearly, completing analysis for a 1,000-student cohort in ~38 ms, which remains well within interactive thresholds.

TABLE VII. RUNTIME (MILLISECONDS, SINGLE RUN)

Students	Grading + rank	Risk scoring
100	3.7	0.8
1,000	31.1	7.3
10,000	217.6	41.5

I. Detection Lead Time

Operating on post-semester cadences, we evaluated temporal detection lead times by replaying the seeded cohort across longitudinal terms, constraining each evaluation solely to historically available records (Table VIII). Across the eight individuals surfaced by either paradigm, the baseline completely misses five. ARMS identifies S2005 as vulnerable as early as Semester 1 due to persistent low attendance, whereas the baseline remains inactive across all eight semesters. S2003 triggers an ARMS alert by Semester 6 once cumulative deficits in both attendance and academic performance coalesce. Conversely, the baseline proves earlier or exclusive only for S2008 (Semester 3) and S2009 (Semester 5), reflecting the heuristic's sensitivity to repeated course failures. S2004 exhibits an initial alert in Semester 1 following an uncharacteristic initial deficit, which naturally de-escalates as cumulative performance recovers—an ephemeral trajectory contextualized by the positive trend driver. Fig. 6 demonstrates the empirical necessity of the override mechanism: because neither S2003 nor S2007 crosses the numerical Medium threshold of 35, their identification depends entirely on attendance breaching the statutory 75% threshold.

TABLE VIII. FIRST SEMESTER FLAGGED (SEEDED COHORT)

ID	ARMS	Baseline	Final ARMS level
S2002	1	1	High
S2003	6	—	Medium
S2004	1	—	Low (cleared)
S2005	1	—	High
S2007	3	—	Medium
S2008	—	3	Low
S2009	—	5	Low
S2010	1	—	Medium

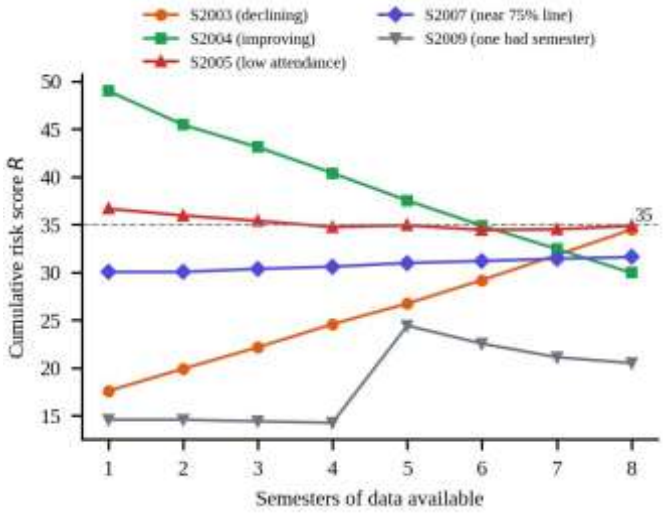


Fig. 6. Cumulative risk score by semester for five seeded students. Dashed line: Medium threshold.

VII. DISCUSSION AND LIMITATIONS

These empirical findings substantiate a focused claim: synthesizing attendance records with academic performance within a single transparent index identifies vulnerable students that standard failure-tally heuristics fail to detect, while matching the discriminative ranking capacity of standard classifiers on synthetic data. However, these results do not establish real-world predictive validity. Because the simulation samples target labels from identical latent constructs as its feature set, all absolute metrics inherently reflect this underlying dependency. Furthermore, our ablation demonstrates that the marginal utility of attendance hinges heavily on its empirical correlation with academic marks. The case cohort similarly reflects ten hand-crafted archetypes rather than field observations, meaning our proposed weights and classification cutoffs remain operational heuristics until calibrated against institution-specific historical outcomes.

A. Implications for Practice

These findings offer three concrete takeaways for deployment:

- **Policy-Aligned Overrides:** The attendance override serves as the critical early indicator for gradually disengaging students; institutions must configure the 65% and 75% thresholds to reflect their specific statutory examination eligibility criteria rather than adopting our values directly.
- **Capacity-Driven Cutoffs:** Operational decision thresholds should balance counseling availability against false alarms; the threshold sensitivity sweep highlights a direct operational trade-off between maximizing recall and overloading advising staff.
- **Actionable Interpretability:** Diagnostic risk drivers provide necessary pedagogical context; an alert indicating "low attendance, declining trend" offers an actionable basis for mentoring, whereas an opaque numerical value generates skepticism.

B. Recommended Calibration Protocol

Institutions should calibrate ARMS against historical institutional cohorts prior to operational adoption. We recommend the following calibration sequence, which is directly codified in the open simulation pipeline for synthetic validation:

- Curate an anonymized administrative corpus comprising assessment marks, attendance logs, and verifiable academic outcomes (such as formal course failure, cumulative backlogs, or exam disqualification), subject to appropriate institutional ethics and privacy approvals.
- Partition cohorts chronologically by academic year—reserving earlier terms for model fitting and subsequent terms for evaluation—to prevent temporal data leakage.
  - Optimize the performance weighting factor w_p alongside classification cutoffs on the historic training cohort, benchmarking discriminative capacity against standard baselines (e.g., logistic regression and tree ensembles) via AUC, precision, and recall metrics.
  - Establish operational decision thresholds calibrated to available academic advising capacity, explicitly communicating anticipated false-positive rates to counseling personnel.

- Re-estimate model parameters annually to account for concept drift in departmental grading criteria and student attendance patterns.

Recalibrate model coefficients and operational thresholds on an annual cycle to mitigate concept drift arising from shifts in departmental grading standards or baseline student attendance behaviors.

C. Limitations

Threats to validity. Several methodological constraints bound these findings:

- **Construct Validity:** Defining student vulnerability strictly through end-semester course failure captures only a subset of academic distress; alternative criteria such as progressive backlog accumulation, formal attrition, and examination debarment reflect distinct risk mechanisms.
- **Internal Validity:** The simulation couples observable features to outcome labels through shared latent variables by construction, inherently constraining synthetic independence.
- **External Validity:** The evaluation models a three-course curriculum within a single department's evaluation scheme and reflects Indian regulatory attendance benchmarks, which may not generalize across differing institutional policies or academic structures.

System limitations. The scoring formulation currently omits subject-level failure counts, which can lead to under-flagging individuals with persistent backlog accumulation. In addition, the two-point trend detection mechanism—relying solely on first-versus-last recorded values—overlooks intermediate performance volatility. Operationally, students registered via spreadsheet imports are not automatically provisioned with user authentication credentials, the system lacks an automated password-reset flow for instructors, and deployment on free-tier infrastructure precludes persistent data storage across restarts.

Ethics. Automated risk classifications carry an inherent hazard of student stigmatization. Consequently, analytical indicators must serve strictly to inform instructor pedagogical oversight rather than automate punitive academic decisions. Student-facing notifications should adopt constructive, supportive framing, and data access controls must strictly adhere to relevant statutory privacy regulations.

VIII. CONCLUSION AND FUTURE WORK

ARMS embeds automated result management alongside an attendance-linked early-warning indicator directly within standard instructor workflows. In synthetic benchmarks, the system identified attendance-driven vulnerabilities overlooked by conventional failure-tally heuristics while achieving ranking performance on par with logistic regression. Our evaluation simultaneously highlighted three operational limitations: a recall-skewed default threshold, an inability to account for historical failure frequency, and diminishing marginal discriminative utility from attendance when performance metrics are already accessible.

Future work will curate anonymized institutional datasets encompassing marks, attendance, and formal pass/fail outcomes under institutional ethical clearance to optimize model weights, refine operational cutoffs, and benchmark against tree-based ensembles. Subsequent iterations will introduce a failure-aware composite heuristic alongside planned enhancements: automated study-plan synthesis, parent alert channels, anomaly detection for grading discrepancies, and expanded analytical administrative views.

REFERENCES

[1] U.S. Department of Education, “Issue brief: Early warning systems.” [Online]. Available: https://www.ed.gov/sites/ed/files/rschstat/eval/highschool/early-warning-systems-brief.pdf

[2] Y.-H. Chang, F.-C. Chen, and C.-I. Lee, “Developing an early warning system with personalized interventions to enhance academic outcomes for at-risk students in Taiwanese higher education,” Educ. Sci., vol. 15, no. 10, Art. no. 1321, Oct. 2025, doi: 10.3390/educsci15101321.

[3] C. Romero and S. Ventura, “Educational data mining: A review of the state of the art,” IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 40, no. 6, pp. 601–618, Nov. 2010.

[4] E. A. Alyahyan and D. Düştegör, “Predicting academic success in higher education: Literature review and best practices,” Int. J. Educ. Technol. High. Educ., vol. 17, Art. no. 3, 2020, doi: 10.1186/s41239-020-0177-7.

[5] K. E. Arnold and M. D. Pistilli, “Course Signals at Purdue: Using learning analytics to increase student success,” in Proc. 2nd Int. Conf. Learn.

Analytics Knowl. (LAK), Vancouver, BC, Canada, 2012, doi: 10.1145/2330601.2330666.

[6] D. Bañeres, M. E. Rodríguez, A. E. Guerrero-Roldán, and A. Karadeniz, “An early warning system to detect at-risk students in online higher education,” Appl. Sci., vol. 10, no. 13, Art. no. 4427, Jun. 2020, doi: 10.3390/app10134427.

[7] L. Cattell and J. Bruch, “Identifying students at risk using prior performance versus a machine learning algorithm,” U.S. Dept. Educ., Inst. Educ. Sci., REL Mid-Atlantic, Rep. REL 2021-126, Sep. 2021.

[8] Pallets Projects, “Flask documentation.” [Online]. Available: https://flask.palletsprojects.com

[9] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[10] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[11] T. Fawcett, “An introduction to ROC analysis,” Pattern Recognit. Lett., vol. 27, no. 8, pp. 861–874, 2006.