IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

An AI Based Dropout Prediction and Counseling System

Pratik Ohol[1], Atharva Pacharne[2], Birendra Pal[3]
Department of BCA, School of Computational Sciences, Faculty of Science and Technology
JSPM University, Pune, Maharashtra, India

Abstract: *Dropout among students is a serious challenge that colleges and universities continue to struggle with, and its consequences go beyond academic failure to affect career outcomes and institutional standings. This paper presents a Dropout Prediction and Counselling System that brings together machine learning and data analytics to flag students who show signs of disengaging before they actually leave. The system draws on a range of inputs — attendance records, test scores, assignment trends, and socio-economic background — to calculate individualized risk scores. Acting on these scores early gives institutions a real chance to step in before a student reaches a point of no return.*

Data collected from students feeds into a structured analysis pipeline that looks for patterns not immediately obvious to instructors. Attendance frequency, assessment outcomes, class participation, and submission habits are all factored in together, because no single indicator on its own tells the full story. When these data points are combined and processed, the model can surface students whose overall trajectory suggests increasing risk — even when their most recent exam score alone would not raise any flags.

What sets this system apart from a plain prediction tool is the counselling layer built on top of it. Rather than just presenting a risk percentage and leaving educators to figure out the next step, the platform recommends concrete interventions — whether that means arranging a mentoring session, flagging a student for financial assistance, or scheduling a follow-up with a counsellor. The goal is to make the output actionable, not just informative.

Keywords: Artificial Intelligence, Machine Learning, Student Dropout Prediction, Educational Data Mining, Predictive Analytics, Counselling System, Random Forest, Explainable AI, Student Retention, Smart Education System

I. INTRODUCTION

Colleges across the country lose a significant portion of their enrolled students before graduation, and the reasons are rarely straightforward. Financial strain, poor academic preparation, personal circumstances, and a general sense of disconnection from institutional life all contribute. For the students involved, leaving without completing a degree has lasting effects on earning potential and career options. For institutions, high dropout rates hurt reputation, funding, and the broader goal of producing qualified graduates.

Advances in data analysis have opened up new possibilities for how institutions track and respond to student struggles. Where a faculty member might notice a student falling behind only after several missed classes and a failed test, a well-designed prediction system can pick up on warning signals much earlier — sometimes across a combination of factors that no single person would connect without computational help. This shift from observation to anticipation is where machine learning genuinely adds value in education.

The system described in this paper was built with one practical aim: give educational institutions a working tool to identify struggling students before their situation deteriorates. It pulls together academic performance data, attendance logs, engagement records, and background information to construct a clearer picture of where each student stands. From this, it produces a risk classification that instructors, counsellors, and administrators can act on.

DOI: 10.48175/IJARSCT-35341

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 6, May 2026 Impact Factor: 8.2

The prediction output is only the starting point. A connected counselling module translates each risk signal into a suggested response — ranging from light monitoring for borderline cases to immediate counsellor assignment for high-risk students. The idea is that the system does not just surface problems; it helps structure the response, saving educators time and ensuring no flagged student slips through the cracks.

Taken together, these capabilities move institutions from a position of reacting to dropout events to one of catching them before they occur. Subtle patterns in attendance trends or score decline rates that might escape notice individually become meaningful signals when analysed as part of a broader student profile.

This system takes that data-driven approach and puts it into practice. Student records across multiple dimensions — academic, behavioural, and socio-economic — are passed through four machine learning models: Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine. Each model contributes to a risk estimate, and the system is designed to make those estimates transparent. Rather than handing educators a black-box score, the platform explains which factors drove a particular prediction, so staff can engage with the result meaningfully.

Prediction alone is not enough to help a student. The counselling component of the system exists precisely because identifying a problem and addressing it are two separate steps. Depending on what the model flags as the primary concern, the system might recommend remedial academic support, a financial aid consultation, psychological counselling, or closer monitoring by the student's assigned mentor. The technical stack — React.js on the frontend, Flask/FastAPI on the backend, and MongoDB/MySQL for storage — is built to handle these workflows reliably at institutional scale.

II. RELATED WORK

Looking at earlier research in this area, a recurring limitation stands out: most existing systems are good at identifying who might drop out but stop short of helping institutions do something about it. Performance dashboards and risk classifiers have been developed in a number of studies, but the step from detection to actual student support tends to be absent or underdeveloped. Closing that gap was one of the central motivations behind this work.

Dropout prediction using machine learning has been explored fairly extensively. Studies have tested Decision Trees, Random Forests, Logistic Regression, and Neural Networks on educational datasets with varying degrees of success. The results are generally encouraging, but a consistent finding is that model accuracy is tightly coupled to the completeness and representativeness of the training data. A model trained on data from one institution may not generalize well to another without recalibration.

Educational data mining platforms have grown more sophisticated over time, with some offering dashboards that track cohort-level trends and flag performance dips.[3] However, these tools generally focus on analysis rather than action. They describe what is happening with student populations but do not connect their findings to any counselling or intervention workflow — leaving faculty to bridge that gap on their own.

Chatbot-based student support tools have gained traction in recent years as a way to provide on-demand guidance without requiring counsellor availability around the clock.[4] These are useful for routine queries but lack the predictive capability needed to identify students who are quietly disengaging. Meanwhile, campus counselling offices continue to operate largely through referral systems and self-referral, with little to no connection to any data-driven early warning process.[5]

What the existing literature reveals is a fragmented landscape: strong prediction tools without counselling capability, counselling platforms without predictive power, and analytics dashboards without actionable outputs. This project was designed with that fragmentation in mind. By building prediction, explainability, and counselling support into one integrated platform, the aim was to produce something institutions could actually use — not just evaluate in a research context.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

III. SYSTEM ARCHITECTURE

A. System Architecture Overview

The system is structured as an end-to-end platform covering everything from data ingestion to intervention management. Rather than treating prediction and student support as separate concerns, the architecture connects them so that a risk flag in one part of the system automatically feeds into the counselling and notification workflows elsewhere. The design goal was to reduce the manual coordination burden on faculty and administrative staff.

Input data reaches the system through several channels. Academic records, attendance registers, assignment logs, behavioral notes from instructors, and family background information are all ingested and normalised before being passed to the prediction engine. Risk scores and trend summaries come out the other side, and the counselling module picks these up to generate intervention suggestions. Each piece of output is then routed to the relevant user — student, teacher, counsellor, or administrator — based on role.

On the technical side, the interface is built in React.js, chosen for its component-based structure which makes it practical to build separate dashboards for different user roles without duplicating code. The backend runs on Flask/FastAPI, handling model inference requests, data routing, and authentication. Storage is split between MySQL for structured records like exam scores and MongoDB for more variable data like counselling notes.

B. User Role Architecture

Four distinct roles define how different actors engage with the system, each with specific functions and information access:

1. Student

Students are the primary focus of the entire platform. Their academic records, attendance history, and submission patterns are the raw material the prediction models work with. On the other side of that process, students can log in and view their own risk profile, check any recommendations the system has generated for them, and see notifications that their mentor or counsellor may have sent through the platform.

2. Teacher

Teachers feed the system with marks, attendance records, and participation data collected during normal course delivery. In return, they get a dashboard view showing risk levels for each of their students. When a student is flagged, teachers can see the specific factors contributing to that risk and, where needed, initiate coordination with the counselling team without any separate communication steps.

3. Counsellor

Counsellors use the platform to review incoming risk flags and build support plans around them. Because the system already highlights what is driving each student's risk score, counsellors can focus on responding rather than gathering information from scratch. The range of interventions they can coordinate through the platform includes one-on-one mentoring, psychological support referrals, study support sessions, and connections to financial aid. Ultimately, the counsellor's role is to translate numbers into action — to take a risk score and turn it into a meaningful conversation with a student who needs help.

4. Administrator

Administrators have the broadest view of the system. Their dashboards aggregate data across departments and programmes, showing overall dropout trends, risk distribution, and intervention outcomes. They also manage user accounts, control system settings, and pull reports that can feed into institutional planning or accreditation submissions.

C. Data Flow Architecture

All data in the system flows through a single centralised pipeline. Students, teachers, and administrators each contribute different types of input that accumulate in the central repository. At regular intervals — or on demand — the machine learning engine processes this repository to update risk predictions. The counselling module picks up those predictions and generates or updates intervention recommendations accordingly. Both the risk scores and the recommendations are made available to users in near real-time through their respective dashboards.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

D. Functional Module Architecture

The Student Management Module stores and manages complete student profiles covering registration details, semester-by-semester academic records, attendance histories, and assignment data. It functions as the primary data source for every other module in the system.

The Prediction and Analytics Module runs four machine learning algorithms — Logistic Regression, Decision Tree, Random Forest, and SVM — against the student dataset and produces risk classifications along with trend reports that teachers and administrators can use to track performance patterns over time.

The Counselling and Recommendation Module takes each risk classification and converts it into a concrete support plan. Based on the specific factors behind a student's risk score, it recommends appropriate actions — tutoring, peer mentoring, psychological support referrals, or financial assistance guidance — and presents these to counsellors and teachers in a form they can act on immediately.

The Administration and Reporting Module covers user account management, access control, and institution-level reporting. Administrators use this module's dashboards to review dropout trends, track retention statistics across departments, and generate reports for governance or accreditation purposes.

IV. DATABASE DESIGN

The database layer uses a hybrid approach combining MySQL and MongoDB. This choice was deliberate: some data in an educational context is highly structured and fits naturally into relational tables — student IDs, marks, attendance counts — while other data, like counselling session notes or behavioral observations, is more variable and better suited to a document store. Using both allows the system to handle each type appropriately rather than forcing everything into a single model.

Each student record in the system holds identifying details alongside the academic and background data the prediction models need — attendance percentages, semester grades, assignment completion rates, and socio-economic indicators. Academic records are stored per subject with full examination history preserved. Whenever the prediction engine runs, its outputs — the risk level, confidence score, algorithm used, and timestamp — are saved as a Prediction document so historical risk trends can be tracked. Counselling records capture session content, the intervention plan agreed upon, and progress notes added over time. Teacher and administrator accounts each carry role-specific credentials that determine what parts of the system they can access.

Cloud hosting was chosen for the deployment environment to take advantage of automatic scaling, managed backups, and the query performance needed when working with data from hundreds or thousands of students simultaneously. The combination of the two database technologies ensures the system handles both the structured side of student data and the more freeform records that come out of counselling interactions.

PRIMARY MONGODB COLLECTIONS AND KEY FIELDS

Collection / Table	Key Fields	Purpose
Students	studentId, fullName, email, attendance, semester, socioEconomicStatus	Stores student personal, academic, and socio-economic information
Academic records	studentId, subjectName, marks, grade, examType, semester	Maintains academic performance and examination history
Predictions	studentId, riskLevel, predictionScore, algorithmUsed, generatedAt	Stores dropout prediction results and analytics
Counselling	studentId, counselorId, counsellingType, remarks, interventionPlan, sessionDate	Maintains counselling sessions and intervention records

DOI: 10.48175/IJARSCT-35341

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

Teachers	teacherId, fullName, department, email, role	Stores teacher and faculty management information
Administrators	adminId, username, password, role, accessLevel	Manages administrator authentication and system access

V. KEY ALGORITHMS

A. Dropout Prediction Algorithm

The prediction engine sits at the core of the system and its job is straightforward: given what is known about a student, estimate how likely they are to disengage or withdraw. It does this by processing multiple features at once — attendance rate, test score trends, assignment submission frequency, classroom participation, and household financial background, among others. The output is a classification into one of three risk tiers: Low, Medium, or High.

Before any model sees the data, it goes through preprocessing — handling missing values, normalising ranges, and selecting the most informative features. Four algorithms are then run: Logistic Regression for a baseline probabilistic estimate, Decision Tree for interpretable rule-based classification, Random Forest for improved accuracy through ensemble learning, and SVM for its effectiveness in high-dimensional feature spaces. The scores each model produces are stored and displayed through role-appropriate dashboards. Random Forest consistently returned the strongest results during testing.

B. Personalized Counselling Recommendation Algorithm

The recommendation engine does not generate one-size-fits-all advice. When a student is flagged, the system looks at which combination of factors drove the risk score and generates suggestions that address those specific issues. A student flagged mainly because of financial indicators gets pointed toward financial aid options. One whose attendance is the primary concern might be assigned to an attendance improvement programme or a mentor check-in.

The range of possible interventions the system can recommend includes academic tutoring, remedial course enrolment, referral to psychological support services, motivational programmes, parental engagement sessions, and financial assistance pathways. Importantly, the system does not treat these as one-off outputs. It continues monitoring the student after an intervention is initiated and updates its recommendations if the situation improves or worsens. This adaptive loop keeps the support relevant rather than becoming outdated after the first session.

VI. IMPLEMENTATION

A. Frontend Stack

The frontend was built with React.js, which made it practical to construct separate dashboard experiences for each user type without redundant code. Students, teachers, counsellors, and administrators each see a version of the interface appropriate to their role. The key features visible through the interface — performance charts, attendance graphs, risk displays, counselling recommendations, and notification feeds — are all rendered in real time as data updates come through from the backend.

HTML, CSS, and JavaScript form the base, with React managing the component lifecycle, state, and routing. The interface was designed to work across screen sizes — a practical requirement given that teachers and administrators may access it from different devices throughout the day. Visualization libraries handle the chart rendering, making trend data and risk distributions easier to read at a glance. All data requests between the frontend and backend go through RESTful API calls.

Each dashboard is deliberately focused on what the relevant user actually needs. A student's view shows their own progress, upcoming recommendations, and any alerts their mentor has sent. Teachers see class-level attendance and risk summaries. Counsellors manage their active caseloads and track ongoing interventions. Administrators get institution-wide aggregates. The separation keeps the interface clean and avoids showing people information they have no practical use for.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

B. Backend Stack

Flask and FastAPI together form the backend layer. Flask handles the standard API routes for data management and user interactions, while FastAPI takes on the heavier prediction workloads where async processing provides a performance advantage. Between them, they coordinate everything the system does behind the scenes: student record management, model inference, counselling record storage, JWT-based authentication, and report generation.

Persistent storage is split between MySQL for structured records and MongoDB for flexible document data, as described earlier. Every authenticated session uses JSON Web Tokens, which means role-based access restrictions are enforced at the API level — a teacher cannot query counselling session details, and a student cannot pull another student's records. The four ML models run as backend services and are invoked on demand when prediction requests come in, either triggered by new data uploads or by manual runs from the admin panel.

Scalability was considered from the beginning of the backend design. Async task queuing allows prediction jobs to run without blocking other requests, and the database queries are optimised to handle large student datasets without noticeable slowdowns. The architecture also exposes integration points for explainability modules, external reporting tools, and SMS or email notification services — making it straightforward to extend the system as needs evolve.

VII. TESTING AND RESULTS

Testing was carried out in several phases covering different aspects of system behaviour. Unit testing was applied to each component individually — registration flows, attendance tracking, prediction generation, counselling output, and individual API endpoints — to confirm that each part behaved correctly in isolation. Once those tests passed, integration testing verified that the components worked together: data entered through the React interface had to reach the database correctly, trigger the right ML model, and return a prediction to the frontend without errors.

End-to-end system tests ran scenarios drawn from real educational workflows, covering all four user roles. Performance testing put the system under load with large batches of student records being processed simultaneously; response times stayed within acceptable limits throughout. Security testing specifically probed the JWT access control layer, confirming that users could not access records outside their role permissions. Usability testing assessed the interface on several device types and screen sizes, with navigation and readability holding up consistently across all of them.

In terms of results, all four models produced valid risk classifications across the test dataset. Random Forest was the standout performer, achieving the highest accuracy and most consistent results across different student profiles. The counselling module generated differentiated plans that varied meaningfully based on the factors driving each student's risk — tutoring referrals for academically struggling students, attendance support plans for chronic absentees, financial aid signposting for students with fee-related flags. Administrator dashboards reflected dropout distribution and departmental risk patterns clearly. Taken together, the testing outcomes showed the system works as intended and is ready for deployment in a real institutional setting.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

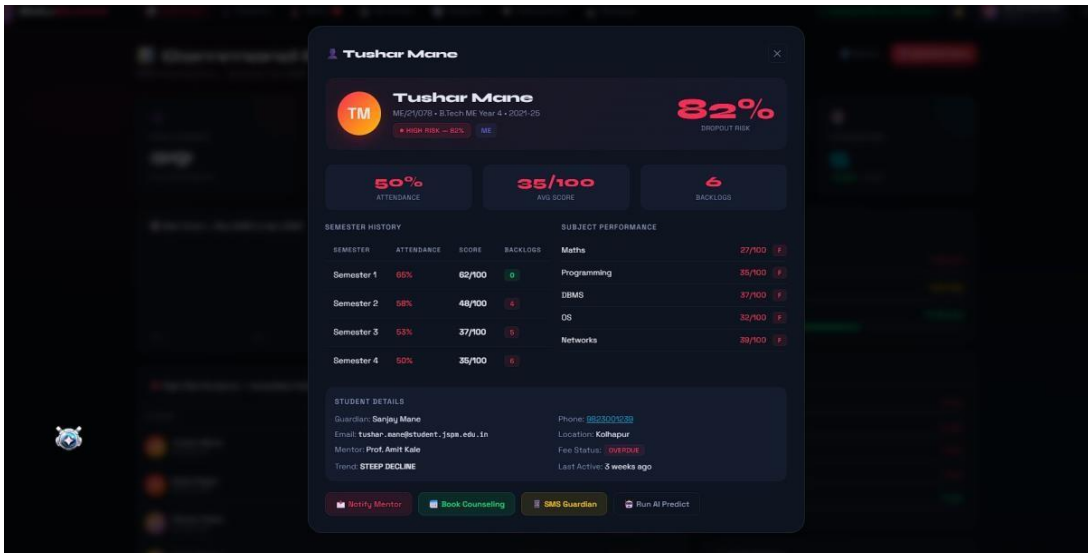

Fig. 1: Student Profile View — Dropout Risk, Semester History, and Subject Performance

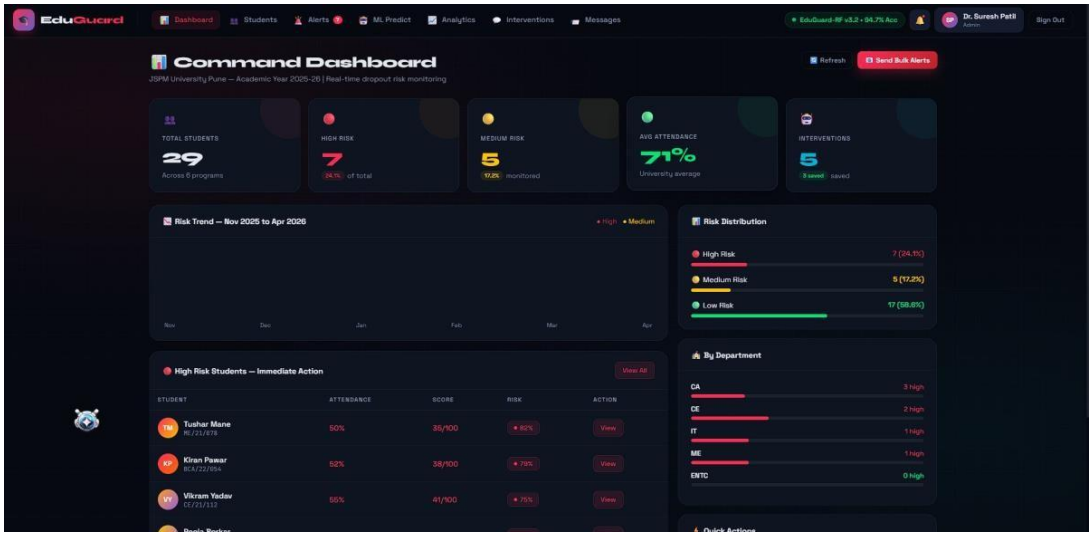

Fig. 2: Command Dashboard — Real-Time Dropout Risk Monitoring and High Risk Student List

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

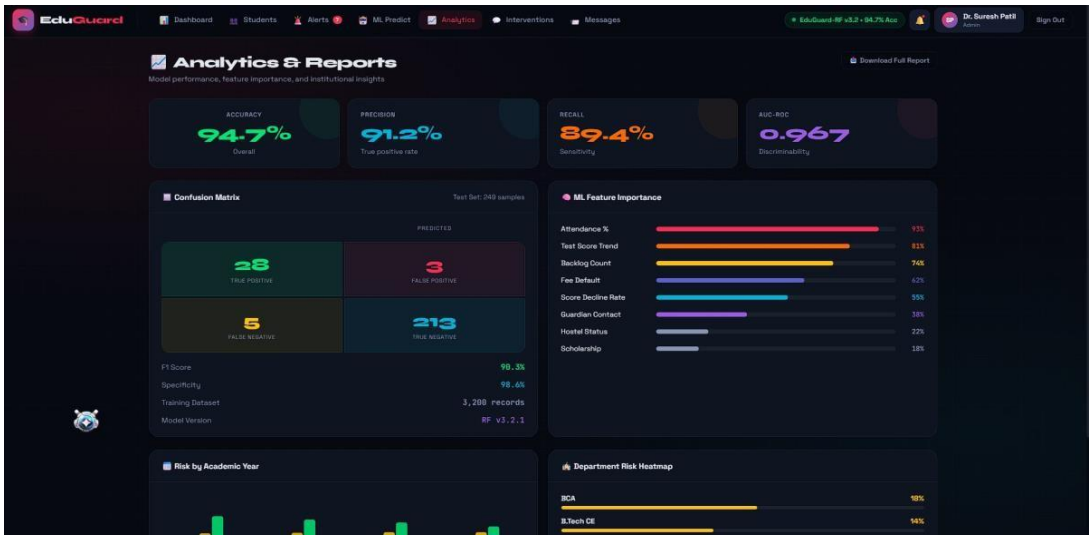

Fig. 3: Analytics and Reports — Model Performance Metrics and Feature Importance

VIII. SDG ALIGNMENT AND SOCIETAL IMPACT

This project connects naturally to the United Nations Sustainable Development Goals, most directly SDG 4 on Quality Education. When institutions can identify students who are at risk of dropping out and respond before the situation deteriorates, retention rates improve and more students complete their programmes. That is a straightforward contribution to the goal of inclusive, equitable, quality education — not just as an aspiration but as a measurable outcome.

SDG 10 — Reduced Inequalities — is also relevant here. Students from lower-income backgrounds or families without a history of higher education are disproportionately represented among those who drop out. By factoring socio-economic indicators into the prediction model and routing those students toward financial support or additional guidance, the system does something that generic academic tracking tools do not: it accounts for the full context of a student's situation, not just their exam scores.

The broader social value of reducing student dropout extends outward from the individual. A student who finishes their degree has better employment prospects, higher lifetime earnings, and greater capacity to contribute to their community. Institutions with stronger retention rates produce a better-trained graduate workforce and build credibility that supports further growth. And when institutions use transparent, data-informed processes to support students, it reflects a standard of accountability in educational governance that benefits everyone — students, staff, and policymakers alike.

IX. LIMITATIONS AND FUTURE SCOPE

No prediction system is better than the data it learns from, and this one is no different. Incomplete records, inconsistently entered attendance data, or historical biases in how students have been assessed all affect model accuracy. The system currently works with what institutions formally collect — grades, attendance, assignment records — which means it cannot easily account for things that happen outside the classroom: a family crisis, a mental health episode, or a financial shock that arrives mid-semester. These factors often drive dropout decisions but rarely appear in institutional databases.

Maintaining the system also requires consistent effort. Keeping training data fresh, updating models as student populations evolve, and ensuring stable server infrastructure are all ongoing commitments that some institutions — particularly smaller or lower-resourced ones — may find difficult to sustain.

DOI: 10.48175/IJARSCT-35341

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 6, May 2026 Impact Factor: 8.2

There are several directions in which this work could grow. Deep learning models would likely handle complex, multi-dimensional risk profiles better than the current ensemble approach, especially for students whose disengagement is gradual and hard to isolate to a single factor. Connecting to live learning management system data — Moodle or Canvas activity logs, for instance — would give the prediction engine earlier and richer signals. Sentiment analysis of student-submitted text could add a window into emotional wellbeing that marks and attendance figures cannot provide. A dedicated mobile interface would make the system more accessible to students who primarily use smartphones. Each of these represents a practical next step rather than a speculative future feature.

X. CONCLUSION

This paper has described a system built to shift how educational institutions respond to student dropout — from reacting after the fact to intervening before disengagement becomes irreversible. Bringing prediction, counselling support, and explainability together in one platform produces something more useful than any of those elements on their own. The combination is the point: a risk score without a counselling response is just a number, and counselling without data-informed targeting is inefficient.

The testing phase validated the system across its main functions. Risk classification was accurate, with Random Forest outperforming the other three models. The counselling module produced support plans that varied sensibly by risk driver rather than defaulting to generic suggestions. Administrative dashboards gave a clear institution-level view of dropout risk distribution. Together, these results confirm that the system works in practice and not just in principle.

What this work points toward is a broader shift in how institutions think about student welfare — from passive monitoring to active support, and from gut-feeling referrals to data-informed decisions. Systems like this one are most valuable not for the technology they contain but for the outcomes they enable: more students finishing their programmes, fewer falling through the cracks, and institutions that can demonstrate they are genuinely invested in the success of every student they enrol.

XI. ACKNOWLEDGMENT

The authors express sincere gratitude to Mrs. Sheetal Sutturwar for expert technical supervision throughout the project; to Dr. Anita Pisote, Project Coordinator, and Dr. Santosh Gaikwad, Programme Coordinator, BCA Department, for their guidance and encouragement; and to Prof. G. A. Patil, Director, and Prof. R. S. Deshpande, Dean, School of Computational Sciences, JSPM University, Pune, for providing the institutional support and facilities that made this work possible.

REFERENCES

1. T. M. Hussain, W. Zhu, W. Zhang, and S. M. Abidi, "Student dropout prediction in higher education using machine learning algorithms," International Journal of Advanced Computer Science and Applications, vol. 10, no. 2, pp. 457–465, 2019.
2. C. Romero and S. Ventura, "Educational Data Mining: A review of the state of the art," IEEE Transactions on Systems, Man, and Cybernetics, vol. 40, no. 6, pp. 601–618, 2010.
3. J. Xu and K. H. Jaggars, "Performance gaps between online and face-to-face courses: Differences across types of students and academic subject areas," The Journal of Higher Education, vol. 85, no. 5, pp. 633–659, 2014.
4. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Data Mining: Practical Machine Learning Tools and Techniques, 4th ed., Morgan Kaufmann, 2016.
5. F. Marbouti, H. A. Diefes-Dux, and K. Madhavan, "Models for early prediction of at-risk students in a course using standards-based grading," Computers & Education, vol. 103, pp. 1–15, 2016.
6. S. Kotsiantis, C. Pierrakeas, and P. Pintelas, "Predicting students' performance in distance learning using machine learning techniques," Applied Artificial Intelligence, vol. 18, no. 5, pp. 411–426, 2004.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

7. UNESCO, Global Education Monitoring Report 2023, UNESCO Publishing, Paris, 2023. [Online]. Available: https://www.unesco.org
8. React Development Team, "React: JavaScript Library for User Interfaces," 2024. [Online]. Available: https://react.dev
9. Flask Development Team, "Flask Documentation," 2024. [Online]. Available: https://flask.palletsprojects.com
10. FastAPI Contributors, "FastAPI Framework Documentation," 2024. [Online]. Available: https://fastapi.tiangolo.com
11. MongoDB, Inc., "MongoDB Atlas Documentation," 2024. [Online]. Available: https://www.mongodb.com/docs/atlas
12. Oracle Corporation, "MySQL Documentation," 2024. [Online]. Available: https://dev.mysql.com/doc
13. Scikit-learn Developers, "Scikit-learn: Machine Learning in Python," 2024. [Online]. Available: https://scikitlearn.org
14. Auth0 by Okta, "JSON Web Token Introduction," 2024. [Online]. Available: https://jwt.io/introduction
15. Ministry of Education, Government of India, National Education Policy 2020, New Delhi, 2020. [Online]. Available: https://www.education.gov.in