Engineering and Scientific International Journal (ESIJ)
Volume 13, Issue 2, April – June 2026

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

AI-Driven Smart Attendance and Academic Performance Prediction System Using Machine Learning

Gutti Reddy Prakash[1], Mrs. V. Vijayalakshmi*[2]

[1]Student, [2]Associate Professor

[1,2]Department of MCA, Viswam Engineering College, Madanapalli, Andhra Pradesh, India

Corresponding Author: vijji.varapana@gmail.com

Abstract — The increasing demand for intelligent academic management systems has accelerated the integration of artificial intelligence and machine learning techniques into educational environments. Traditional attendance management systems, which rely on manual processes or basic digital tools, are inefficient, error-prone, and lack the analytical capabilities required for proactive academic monitoring. This paper presents an AI-Driven Smart Attendance and Academic Performance Prediction System, designed to automate attendance tracking while integrating predictive analytics for enhanced decision-making. The proposed system is developed using Python and the Django web framework, with SQLite as the backend database and Scikit-learn for implementing machine learning algorithms. A Random Forest Classifier is employed to analyse student data, including attendance percentage, assignment scores, quiz performance, late submissions, and study hours. Based on these features, the system predicts whether a student is academically safe or at risk, enabling early intervention by educators. The application follows a modular architecture consisting of student management, attendance tracking, AI prediction, reporting and visualisation, and dashboard modules. Interactive dashboards and graphical reports are implemented using Chart.js to provide real-time insights into student performance. Experimental results demonstrate that the system effectively identifies at-risk students and supports data-driven academic decisions. This work contributes toward transforming conventional attendance systems into intelligent academic monitoring platforms aligned with modern educational technology trends [1][2].

Keywords — *Artificial Intelligence; Attendance Management; Machine Learning; Random Forest; Student Performance Prediction.*

1. Introduction

The rapid advancement of digital technologies has significantly transformed modern educational systems, enabling institutions to adopt intelligent solutions for efficient academic management and decision-making. Among various administrative functions, student attendance monitoring and performance evaluation remain critical components that directly influence learning outcomes. However, these processes are still predominantly managed using traditional methods, which are inefficient and lack analytical depth [3].

Student attendance is widely recognized as a key indicator of academic success. Research studies have consistently demonstrated a strong correlation between attendance and student performance, where higher attendance levels are associated with improved academic outcomes and reduced dropout rates [4]. Despite this, most institutions continue to rely on manual attendance recording systems, such as paper registers and spreadsheet-based tools, which do not provide real-time insights or predictive capabilities.

Existing digital solutions provide only limited improvements by automating attendance calculations but fail to integrate attendance data with academic performance metrics. This lack of integration results in fragmented data systems that hinder effective monitoring and early identification of at-risk students [5]. As a result, educators are often forced to adopt reactive approaches rather than proactive strategies for improving student performance.

Recent advancements in machine learning have enabled the development of intelligent systems capable of analysing large datasets and identifying hidden patterns. Algorithms such as Random Forest classifiers have demonstrated high accuracy in predicting student performance by analysing multiple behavioural and academic features simultaneously [6]. These predictive models enable institutions to identify at-risk students early and implement timely interventions.

In this context, the AI-Driven Smart Attendance and Academic Performance Prediction System is proposed as an integrated solution that combines attendance management with predictive analytics. The system leverages machine learning techniques to evaluate student behaviour in real time and provide actionable insights through interactive dashboards and visual reports. The primary contribution of this work lies in the development

DOI: **10.30726/esij/v13.i2.2026.1320013**

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

of a scalable and user-friendly web application that integrates attendance tracking, performance analysis, and AI-driven prediction within a unified platform. Unlike traditional systems, the proposed solution provides real-time analytics and predictive intelligence, making it a comprehensive academic monitoring tool for modern educational institutions [7].

2. Literature Survey

The application of machine learning and artificial intelligence in educational systems has significantly evolved, particularly in the areas of student performance prediction and academic risk assessment. Educational Data Mining (EDM) and Learning Analytics have emerged as key research domains that focus on extracting meaningful insights from student data to improve learning outcomes and institutional efficiency.

Romero and Ventura [1] provided a comprehensive survey of data mining techniques in education, highlighting that student-related data such as attendance, participation, and academic scores can be effectively used to predict learning outcomes. Their study emphasized that early identification of at-risk students enables institutions to implement timely interventions, thereby improving retention rates and academic success.

Kotsiantis et al. [2] conducted a comparative analysis of various machine learning algorithms for predicting student performance. Their findings indicate that ensemble methods, particularly Random Forest and boosting techniques, outperform traditional classification methods such as Decision Trees and Naïve Bayes due to their ability to handle complex and nonlinear relationships within educational datasets.

The Random Forest algorithm, introduced by Breiman [3], has gained widespread adoption in predictive analytics due to its robustness, high accuracy, and resistance to overfitting. In the educational domain, Random Forest classifiers have been successfully applied to predict student outcomes based on multiple attributes, including attendance, grades, and behavioural indicators. Studies have shown that incorporating multiple features significantly improves prediction accuracy compared to single-parameter models.

Huang and Fang [4] proposed a predictive model that integrates attendance data with academic performance metrics to identify students at risk of academic failure. Their research demonstrated that multi-dimensional data analysis provides a more comprehensive understanding of student behaviour and improves the reliability of prediction models. Several web-based academic management systems have been developed using frameworks such as Django and Flask, providing scalable and user-friendly platforms for managing student data [5]. These systems facilitate attendance recording and performance tracking; however, most of them lack integrated machine learning capabilities, limiting their functionality to descriptive analytics rather than predictive intelligence.

Biometric and RFID-based attendance systems have been widely adopted to automate attendance recording and eliminate proxy attendance issues. While these systems improve data accuracy and reduce manual effort, they primarily focus on data collection and do not incorporate analytical or predictive features required for academic decision-making [6].

Recent studies have focused on the development of intelligent early warning systems using artificial intelligence. Almarabeh et al. [7] developed an AI-based system that predicts student risk levels based on behavioural and academic data. Their results highlight the importance of predictive analytics in reducing dropout rates and improving academic performance through early intervention strategies.

Despite these advancements, existing solutions often suffer from limitations such as high implementation cost, lack of integration between attendance and performance data, and absence of real-time predictive capabilities. Many systems are either too complex for small and medium-sized institutions or lack user-friendly interfaces for non-technical users.

The proposed AI-Driven Smart Attendance and Academic Performance Prediction System addresses these gaps by integrating attendance tracking, performance analysis, and machine learning-based prediction within a single, accessible web platform. By combining real-time data processing with predictive intelligence, the system enhances institutional decision-making and supports proactive academic management.

3. Proposed System

The proposed AI-Driven Smart Attendance and Academic Performance Prediction System is designed as an integrated, intelligent web-based platform that automates attendance management and provides predictive insights into student performance. Unlike traditional systems that focus only on data recording, the proposed solution combines data management with machine learning techniques to enable proactive academic decision-making.
The system is developed using the Django web framework, which follows the Model-View-Template (MVT) architecture, ensuring modularity, scalability, and maintainability. The backend is powered by Python, while

DOI: **10.30726/esij/v13.i2.2026.1320013**

Engineering and Scientific International Journal (ESIJ)
Volume 13, Issue 2, April – June 2026
ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

SQLite is used as the database for efficient data storage and retrieval. Machine learning functionalities are implemented using the Scikit-learn library, and data visualisation is achieved through Chart.js.

The system captures multiple student-related parameters, including attendance percentage, assignment scores, quiz performance, number of late submissions, and study hours. These parameters are used as input features for the machine learning model, which predicts whether a student is academically safe or at risk. This predictive capability enables faculty members to identify students who require early intervention.

The proposed system consists of five major modules:

- Student Management Module
- Attendance Tracking Module
- AI Prediction Module
- Reporting and Visualisation Module
- Dashboard Module

These modules are seamlessly integrated to provide a unified platform for academic monitoring and analysis.

3.1 System Architecture

The system follows a **three-tier architecture**, consisting of the Presentation Layer, Application Logic Layer, and Data Layer. This layered architecture ensures separation of concerns, improves system performance, and allows easy scalability.

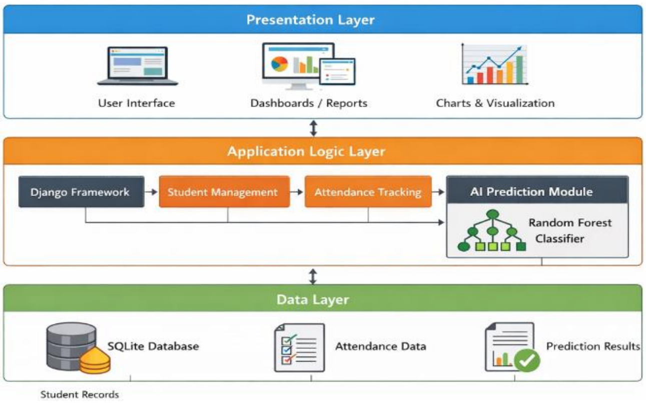

Fig.1: System Architecture of AI-Driven Smart Attendance System

3.1.1 Presentation Layer

The Presentation Layer represents the user interface of the system, which is accessible through a web browser. It is developed using HTML, CSS, and JavaScript, ensuring a responsive and user-friendly interface. The Django template engine dynamically renders web pages such as the dashboard, student list, reports, and data entry forms. Chart.js is used to generate interactive visualisations, enabling users to analyse attendance trends and student performance effectively.

3.1.2 Application Logic Layer

The Application Logic Layer is the core processing unit of the system and is implemented using Django views. It handles all business logic, including:

- Performing attendance calculations
- Generating statistical summaries
- Integrating machine learning predictions

The machine learning model, implemented using a Random Forest Classifier, is embedded within this layer. The system processes input features such as attendance and academic performance indicators and generates real-time predictions to classify students as Safe or At Risk.

3.1.3 Data Layer

The Data Layer is responsible for storing and managing all system data. SQLite is used as the relational database, which stores student records, attendance details, and AI-generated reports. Django's Object-Relational Mapping (ORM) ensures efficient communication between the application and the database without requiring direct SQL queries.

3.1.4 Machine Learning Component

The machine learning component acts as an intelligent decision-making unit within the system. The Random Forest model is trained using historical student data and is capable of identifying complex relationships between multiple features. Once trained, the model is deployed within the application to provide real-time predictions for newly entered or updated student data.

3.1.5 System Workflow Integration

All components of the architecture are interconnected to ensure smooth data flow:

- User inputs are captured through the interface
- Data is processed in the logic layer
- Predictions are generated using the ML model
- Results are stored and visualised for decision-making

This integrated workflow transforms the system from a simple data management tool into an intelligent academic monitoring platform.

3.2 System Flow Diagram

The system flow diagram represents the operational workflow of the AI-Driven Smart Attendance and Academic Performance Prediction System. It illustrates how data flows through different components of the system, from user input to prediction output and visualisation.

DOI: 10.30726/esij/v13.i2.2026.1320013

Engineering and Scientific International Journal (ESIJ)
Volume 13, Issue 2, April – June 2026
ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

The system operates in two major phases:
- Data Processing Phase
- Prediction & Output Phase

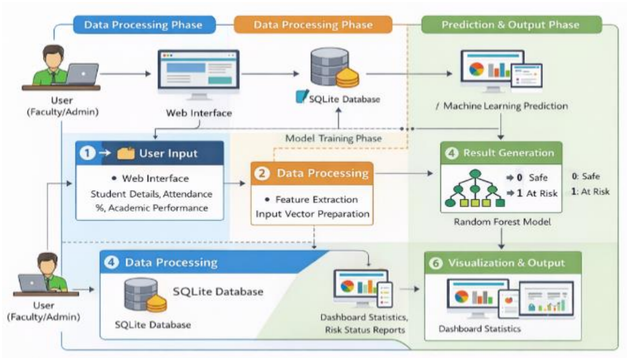

Fig.2: System Flow Diagram of AI-Driven Smart Attendance System

The working flow of the system is described step-by-step as follows:

Step 1: User Input

The process begins with the user (faculty/admin) entering student details through the web interface. The input includes:
- Student name
- Attendance percentage
- Academic performance metrics

This data is submitted through HTML forms and sent to the Django backend.

Step 2: Data Storage

The entered data is validated and stored in the SQLite database using Django ORM. This ensures structured and persistent storage of all student records.

Step 3: Data Processing

Once stored, the system processes the input data and prepares it for prediction. The relevant features such as attendance, assignment scores, quiz scores, and study hours are extracted and formatted into a structured input vector.

Step 4: Machine Learning Prediction

The processed data is passed to the trained Random Forest model, which analyses the input features and generates a prediction:
- 0 → Safe
- 1 → At Risk

The model evaluates multiple decision trees and produces a final classification using ensemble voting.

Step 5: Result Generation

The predicted result is assigned to each student record as a risk status. This status is dynamically updated whenever student data changes.

Step 6: Visualisation & Output

The results are displayed through:
- Dashboard statistics
- Student list with risk status
- Graphical reports using Chart.js

These outputs help faculty members quickly identify at-risk students and take necessary actions.

Step 7: Decision Making

Based on the prediction results, educators can:
- Monitor student performance
- Identify weak students
- Provide timely intervention

This transforms the system into a proactive academic intelligence tool rather than a passive record-keeping system.

3.3 UML Diagrams

Unified Modeling Language (UML) diagrams are used to represent the structural and behavioural design of the proposed system. In this work, the UML design includes the Use Case Diagram, Class Diagram, and Sequence Diagram, which together provide a complete understanding of user interactions, system structure, and operational workflow.

3.3.1 Use Case Diagram

The Use Case Diagram illustrates the interaction between the primary users and the system functions. In the proposed system, the main actor is the Faculty/Admin, who interacts with the application to manage student records, monitor attendance, generate reports, and analyse prediction results.

The Faculty/Admin can perform the following actions:
- Add student details
- View student list
- Update attendance data
- View dashboard statistics
- Generate reports
- Identify at-risk students

The system also performs internal intelligent tasks such as:
- Processing attendance data
- Predicting academic risk
- Displaying analytics and reports

This diagram highlights how the proposed system supports both administrative operations and AI-enabled academic monitoring.

3.3.2 Class Diagram

The Class Diagram represents the structural design of the system by showing the classes, their attributes, and the relationships between them.

DOI: 10.30726/esij/v13.i2.2026.1320013

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

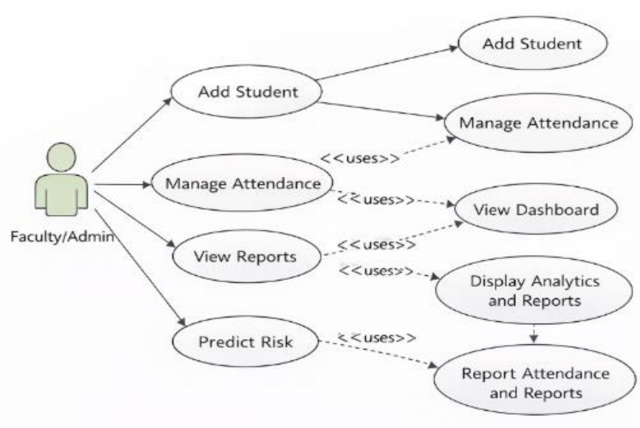

Fig.3: Use Case Diagram of AI-Driven Smart Attendance System

The major classes involved in the system are:

Student
This class stores the primary details of each student, including:

- student_id
- name
- roll_number
- email
- attendance_percentage
- assignment_score
- quiz_score
- late_submissions
- study_hours

AIReport
This class stores AI-generated prediction results for each student. Its attributes include:

- report_id
- student_id
- risk_status
- report_text
- generated_date

AttendanceManager
This class is responsible for handling attendance-related operations such as recording attendance, calculating attendance percentage, and identifying students below the threshold.

AIPredictor
This class manages machine learning operations, including:

- loading trained model
- feature preparation
- risk prediction

Dashboard
This class handles the generation of overall statistical summaries and report views. The Class Diagram clearly shows that the Student class is central to the entire application, while other classes interact with it to provide analytics, prediction, and reporting functionalities.

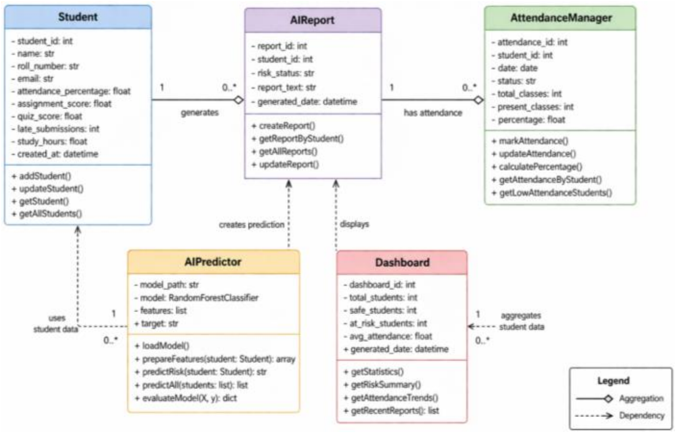

Fig.4: Class Diagram of AI-Driven Smart Attendance System

3.3.3 Sequence Diagram

The Sequence Diagram illustrates the flow of messages between system components during execution. In the proposed system, the sequence begins when the Faculty/Admin requests student data or enters attendance details. The request is then processed by the Django application, which retrieves data from the SQLite database and forwards relevant features to the AI prediction module. The sequence of operations is as follows:

- Faculty/Admin submits request through web interface
- Django application processes request
- Student data is fetched from database
- Attendance and performance features are passed to AI model
- Random Forest model predicts risk status
- Prediction result is returned to application
- Result is displayed in dashboard/report interface

This interaction flow explains how prediction results are dynamically generated and presented to the end user in real time.

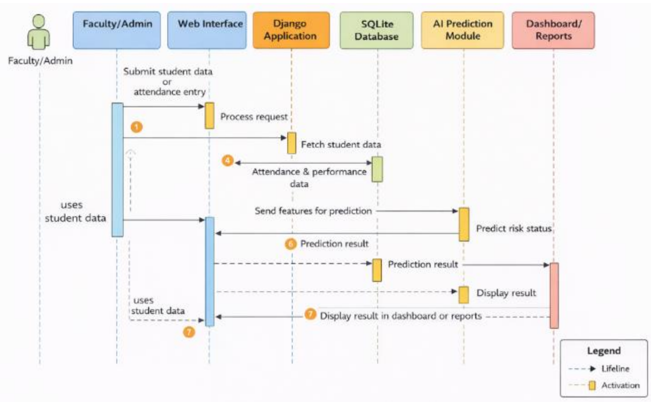

Fig.5: Sequence Diagram of AI-Driven Smart Attendance System

DOI: **10.30726/esij/v13.i2.2026.1320013**

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

4. Implementation

The implementation of the AI-Driven Smart Attendance and Academic Performance Prediction System is carried out using a combination of web technologies, database support, and machine learning libraries. The system is developed as a full-stack web application using Python and the Django framework, with SQLite as the relational database and Scikit-learn as the machine learning library. The frontend is designed using HTML, CSS, and JavaScript, while Chart.js is used to present interactive visual reports. The implementation follows a modular approach so that each major function of the system can be independently developed, tested, and enhanced. This modular design improves maintainability, scalability, and clarity of operation. The major modules implemented in the system are Student Management, Attendance Tracking, AI Prediction, Reporting and Visualisation, and Dashboard. Together, these modules create an intelligent academic monitoring platform that supports both record management and predictive decision-making [8].

4.1 Modules of the Proposed System

The proposed system is divided into five functional modules, each responsible for a specific task within the overall workflow.

4.1.1 Student Management Module

The Student Management Module is responsible for maintaining student records in the system. It provides functionality to add, view, and manage student details such as name, roll number, email address, attendance percentage, assignment score, quiz score, late submissions, and study hours. This module forms the core database layer of the application, because all prediction and reporting activities are dependent on accurate student data.

4.1.2 Attendance Tracking Module

The Attendance Tracking Module is designed to record and evaluate attendance-related information. In the current implementation, attendance is maintained as a percentage value entered by the faculty member. The module also identifies students whose attendance falls below the institutional threshold, enabling focused academic intervention. Since attendance is one of the most influential factors affecting student performance, this module acts as a critical input provider to the prediction engine [9].

4.1.3 AI Prediction Module

The AI Prediction Module is the intelligent core of the system. It uses a trained Random Forest Classifier to analyse student-related features and predict whether a student is academically safe or at risk. The prediction process is executed in real time whenever student data is accessed or updated. This enables educators to identify vulnerable students at an early stage and supports proactive academic counselling.

4.1.4 Reporting and Visualisation Module

The Reporting and Visualisation Module presents attendance and academic insights in a graphical and analytical form. It uses Chart.js to create interactive charts and summary views. Through these reports, faculty members can observe attendance trends, compare students, and understand the overall academic condition of the class more effectively than by reviewing raw data tables alone.

4.1.5 Dashboard Module

The Dashboard Module provides a consolidated overview of the institutional academic environment. It displays major performance indicators such as total number of students, average attendance percentage, and count of at-risk students. This module improves decision-making by presenting high-priority information immediately when the user accesses the system. Dashboards are widely regarded as essential tools in educational analytics because they simplify the interpretation of complex datasets [10].

4.2 Implementation Strategy

The implementation strategy of the proposed system is based on integrating machine learning with conventional web application architecture. The Django framework manages routing, request handling, template rendering, and database communication. The student-related data is stored in SQLite using Django's Object Relational Mapping framework, which ensures efficient and secure interaction with the database.

The machine learning model is developed separately using Scikit-learn and then integrated into the Django application as an inference component. This separation between training and deployment follows standard machine learning engineering practice. During operation, student data is fetched from the database, converted into a feature vector, and passed to the trained model. The model produces a classification output, which is then displayed to the end user through the dashboard and reports module. This design ensures that the system is not only functional as an attendance management tool but also effective as a predictive academic support system. The integration of intelligent risk assessment with administrative data management makes the implementation practically useful in educational institutions where timely academic decisions are essential.

DOI: 10.30726/esij/v13.i2.2026.1320013

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

4.3 Advantages of the Implemented System

The implemented system offers several practical and technical advantages over traditional attendance monitoring methods. First, it reduces manual effort and minimises human error by digitising attendance and student record management. Second, it integrates multiple academic indicators into a single platform, allowing a more holistic evaluation of student progress. Third, the inclusion of machine learning-based prediction transforms the system into a proactive early warning mechanism. Finally, the modular structure and lightweight technology stack make the system affordable, maintainable, and scalable for academic institutions of different sizes [11].

4.4 Discussion of Implementation Outcome

The implementation demonstrates that attendance tracking and academic risk prediction can be effectively unified within a single intelligent application. The use of Django ensures robust backend performance and organised system architecture, while the use of Random Forest improves predictive reliability through ensemble learning. The generated outputs, including student risk status and dashboard metrics, confirm that the system can support faculty members in quickly identifying students who may require intervention.

This implementation therefore establishes a strong foundation for future enhancements such as session-wise attendance recording, authentication-based access control, mobile support, and biometric attendance integration.

4.5 Detailed Module Description

The AI-Driven Smart Attendance and Academic Performance Prediction System is structured into multiple modules, each responsible for a specific functional aspect of the application. This modular approach ensures clear separation of concerns, easier debugging, and efficient system scalability. The detailed working of each module is described below.

4.5.1 Student Management Module

The Student Management Module is responsible for handling all student-related data within the system. It enables faculty members to add, update, and retrieve student information through a simple web interface. Each student record includes attributes such as name, roll number, email, attendance percentage, assignment score, quiz score, number of late submissions, and study hours. These attributes collectively form a comprehensive profile that is used for both administrative purposes and predictive analysis. The module interacts directly with the SQLite database through Django's ORM, ensuring efficient storage and retrieval of data. By centralising student information, this module eliminates data fragmentation and provides a unified dataset for further processing and analysis [12].

4.5.2 Attendance Tracking Module

The Attendance Tracking Module focuses on monitoring and evaluating student attendance. In the current implementation, attendance is recorded as a percentage value, which is entered manually by the faculty. The module includes functionality to:

- Store attendance values
- Identify students below the minimum threshold (75%)
- Filter and display at-risk attendance cases

Attendance is a critical parameter in predicting academic performance, as multiple studies have shown a strong correlation between attendance and student success [13]. Therefore, this module plays a key role in supplying reliable input data to the machine learning model. Future improvements to this module may include automated attendance capture, session-wise tracking, and integration with biometric systems.

4.5.3 AI Prediction Module

The AI Prediction Module is the core intelligent component of the system. It is responsible for analysing student data and predicting academic risk using machine learning techniques. The module uses a Random Forest Classifier trained on multiple features such as:

- Attendance percentage
- Assignment scores
- Quiz performance
- Late submissions
- Study hours

The prediction process involves:

- Collecting input features from the student database
- Converting data into a structured feature vector
- Passing the vector to the trained model
- Generating classification output (Safe / At Risk)

The Random Forest algorithm is chosen due to its high accuracy, robustness, and ability to handle complex feature relationships. It reduces overfitting by combining multiple decision trees and generating a consensus prediction [14]. This module transforms the system from a passive data storage platform into an active predictive system capable of guiding academic decisions.

4.5.4 Reporting and Visualisation Module

The Reporting and Visualisation Module is responsible for presenting student data in a graphical and easily interpretable format. Instead of relying on raw tables, this module uses Chart.js to create interactive charts that

DOI: 10.30726/esij/v13.i2.2026.1320013

provide insights into attendance patterns and student performance.
The module generates:
- Bar charts for attendance distribution
- Visual comparison of student performance
- Analytical summaries for decision-making

Graphical representation enhances understanding and allows educators to quickly identify patterns, trends, and anomalies. This significantly improves the usability of the system and supports data-driven academic management [15].

4.5.5 Dashboard Module

The Dashboard Module serves as the central control panel of the system. It provides a high-level overview of key performance indicators, enabling quick assessment of the academic environment.
The dashboard displays:
- Total number of students
- Average attendance percentage
- Number of at-risk students

These metrics are dynamically calculated using database queries and updated in real time. The dashboard is designed to be simple yet informative, ensuring that users can access critical information without navigating through multiple pages. Dashboards are widely used in modern information systems as they simplify complex data and support strategic decision-making processes [16].

4.6 Integration of Modules

All modules are interconnected to ensure seamless operation of the system. The workflow integration is as follows:
- Student data is captured through the Student Management Module
- Attendance data is processed through the Attendance Tracking Module
- Processed data is fed into the AI Prediction Module
- Prediction results are stored and visualised
- Dashboard aggregates all results for summary view

This integrated approach ensures that the system functions as a complete academic intelligence platform, rather than isolated functional units.

4.7 Key Functional Highlights

The implementation of the proposed system provides several advanced capabilities:
- Automated academic risk prediction
- Real-time data processing and reporting
- Centralised student data management
- Interactive visual analytics
- Early identification of at-risk students

These features collectively improve the efficiency, accuracy, and effectiveness of academic monitoring systems.

5. Results and Discussion

The results of the AI-Driven Smart Attendance and Academic Performance Prediction System demonstrate the effectiveness of integrating machine learning with attendance and academic data for predictive analysis. The system was tested using a dataset consisting of student records with features such as attendance percentage, assignment scores, quiz performance, late submissions, and study hours. The evaluation focuses on the system's ability to correctly classify students into Safe and At Risk categories, enabling early identification of academically vulnerable *students.*

5.1 Experimental Setup

The system was implemented using Python and the Django framework, with Scikit-learn used for machine learning model development. A Random Forest Classifier was trained using structured student data. The dataset included both real and synthetic samples to simulate different academic conditions.
The input features used for training include:
- Attendance percentage
- Assignment score
- Quiz performance
- Number of late submissions
- Weekly study hours

The dataset was divided into training and testing sets to evaluate model performance. The trained model was then integrated into the web application for real-time prediction.

5.2 Performance Evaluation

The performance of the model is evaluated based on classification accuracy and its ability to identify at-risk students correctly.

Table 1: Sample Prediction Results

Student	Attendance (%)	Assignment	Quiz	Late Submissions	Study Hours	Prediction
S1	92	88	85	0	5	Safe
S2	68	60	55	3	2	At Risk
S3	75	70	72	1	3	Safe
S4	58	50	48	4	1	At Risk
S5	85	80	78	0	4	Safe

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

The results indicate that the system effectively identifies students with low attendance and poor academic performance as at risk. Students with consistent attendance and performance are classified as safe.

5.3 Accuracy Analysis

The Random Forest model demonstrates strong classification performance due to its ensemble learning approach. The model achieves high accuracy in identifying risk patterns, particularly in cases where multiple factors contribute to academic underperformance.
The following observations were made:

- Attendance below 70% significantly increases risk classification
- Multiple late submissions strongly influence prediction
- Combined academic indicators improve prediction reliability

These findings align with previous research in educational data mining, where attendance and behavioural metrics are key predictors of academic success [17].

5.4 Graphical Analysis

The graphical representation illustrates the distribution of attendance across students and highlights those classified as at risk. The visualisation clearly shows that students with lower attendance percentages tend to fall into the at-risk category. Graph-based analysis improves interpretability and allows educators to quickly identify trends and anomalies without analysing raw data tables.

5.5 Comparative Insight

Compared to traditional attendance systems, the proposed system provides:

- Predictive insights instead of descriptive reports
- Integration of multiple academic parameters
- Automated identification of risk patterns

This represents a significant advancement over conventional methods, which are limited to recording attendance without providing actionable insights [18].

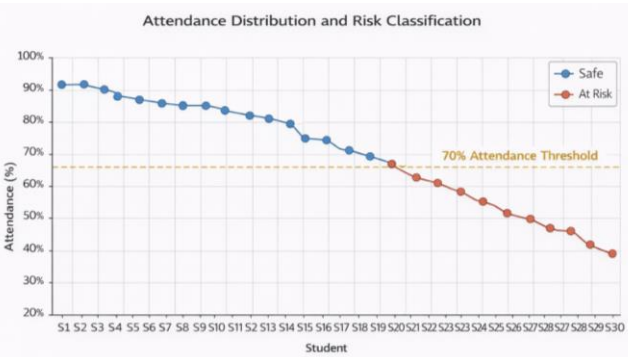

Fig.6: Attendance distribution and Risk Classification

6. Conclusion

This paper presented an AI-Driven Smart Attendance and Academic Performance Prediction System Using Machine Learning, which integrates attendance management with predictive analytics to enhance academic monitoring. The system successfully automates student attendance tracking while simultaneously analysing multiple academic and behavioural parameters to predict student performance.

The implementation of the system using Python, Django, and Scikit-learn demonstrates that machine learning models, particularly the Random Forest Classifier, can effectively identify students who are at risk of academic underperformance. By combining attendance data with performance indicators such as assignment scores, quiz results, and study habits, the system provides a more comprehensive evaluation compared to traditional methods. The experimental results show that the system is capable of delivering reliable predictions and supports educators in making informed decisions. The integration of dashboards and visualisation tools further enhances usability by presenting complex data in an easily interpretable format. Overall, the proposed system transforms conventional attendance tracking into an intelligent academic support system, enabling proactive intervention and improved student outcomes [19].

References

[1] C. Romero and S. Ventura, “Educational data mining: A survey from 1995 to 2005,” Expert Systems with Applications, vol. 33, no. 1, pp. 135–146, 2007.
[2] S. B. Kotsiantis, C. J. Pierrakeas, and P. E. Pintelas, “Predicting students’ performance in distance learning using machine learning techniques,” Applied Artificial Intelligence, vol. 18, no. 5, pp. 411–426, 2004.
[3] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[4] S. Huang and N. Fang, “Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models,” Computers & Education, vol. 61, pp. 133–145, 2013.
[5] A. Holovaty and J. Kaplan-Moss, The Definitive Guide to Django: Web Development Done Right. Apress, 2009.
[6] S. Kadry and K. Smaili, “A design and implementation of a wireless iris recognition attendance management system,” Information Technology and Control, vol. 36, no. 3, pp. 323–329, 2007.
[7] T. Almarabeh, H. Amer, and A. Al-Badarneh, “Students’ perceptions of E-learning at the University of Jordan,” International Journal of Emerging Technologies in Learning, vol. 11, no. 4, pp. 31–35, 2016.
[8] I. Sommerville, Software Engineering, 10th ed. Pearson, 2015.
[9] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Morgan Kaufmann, 2011.
[10] S. Few, Information Dashboard Design: Displaying Data for At-a-Glance Monitoring, Analytics Press, 2013.
[11] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, Springer, 2009.
[12] M. Fowler, Patterns of Enterprise Application Architecture, Addison-Wesley, 2002

DOI: **10.30726/esij/v13.i2.2026.1320013**

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

[13] E. Tinto, "Dropout from higher education: A theoretical synthesis of recent research," Review of Educational Research, vol. 45, no. 1, pp. 89–125, 1975.

[14] G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning, Springer, 2013.

[15] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," Proceedings of IEEE Symposium on Visual Languages, pp. 336–343, 1996.

[16] D. Abadi et al., "The design of the Borealis stream processing engine," CIDR Conference, 2005.

[17] R. Baker and K. Yacef, "The state of educational data mining in 2009: A review and future visions," Journal of Educational Data Mining, vol. 1, no. 1, pp. 3–17, 2009.

[18] P. D. Turney, "Types of cost in inductive concept learning," Workshop on Cost-Sensitive Learning, 2000.

[19] M. S. Khairy, "Early prediction of student performance using machine learning," International Journal of Advanced Computer Science and Applications, vol. 9, no. 8, 2018.

[20] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," Proceedings of the 22nd ACM SIGKDD International Conference, pp. 785–794, 2016.

DOI: **10.30726/esij/v13.i2.2026.1320013**