

AI-Based Knowledge Gap Detection System for Student Performance Analysis Using Predictive Analytical Models

Megha Kumarappa Doddamani¹, Vijaylakshmi H Hunakunti¹, Rashmi K. C.¹, Nageshwari C.¹

¹*Department of Artificial Intelligence and Data Science, Shridevi Institute of Engineering and Technology, Tumakuru, Karnataka, India*

Visvesvaraya Technological University, Belagavi, Karnataka, India

Abstract—A single aggregate exam score hides which topics a student has actually mastered: two students with the same overall percentage may have entirely different strengths and weaknesses. This paper presents an AI-based knowledge gap detection system that analyzes quiz performance at the topic level. Each attempt is linked to a student's roll number and scored automatically, and responses are grouped by topic to compute topic-wise accuracy. Topics below a mastery threshold are flagged as knowledge gaps, and the weakest flagged topic is recommended for practice next. A separate communication module captures a spoken answer through the browser's speech-recognition interface and scores the transcript against a subject-specific keyword bank, adding an indicator of verbal articulation. The system uses a FastAPI backend, a browser-based dashboard and a scikit-learn regression baseline for accuracy prediction, and is evaluated through functional verification, analysis of a 100-student cohort data set and a held-out predictive-modelling experiment. The results show that topic-level analysis exposes gaps that an overall score conceals, while the minimal predictive baseline proves inadequate, which motivates the richer features and adaptive feedback loop proposed as future work.

Keywords—Educational data mining, Knowledge gap detection, Machine learning, Speech-based communication assessment, Student performance analysis.

I. INTRODUCTION

Digital learning environments now generate a steady stream of data about how students work. Learning management systems, online quizzes and digital tests record interaction traces ranging from the correctness of single items to the timing of responses, and these traces hold diagnostic information that one examination score cannot convey. Yet in most classrooms a student's capability is still reduced to a single percentage or grade. Two students who both obtain 70% in a test can have very different mastery profiles: one may handle algorithms confidently but struggle with database normalization, while the other shows exactly the opposite pattern. An aggregate score cannot tell them apart, and a teacher who wishes to support either student has to go back through the answer script manually. Such inspection does not scale beyond a handful of students and becomes harder when a spoken or communication component also forms part of the evaluation.

Educational Data Mining (EDM) and Learning Analytics (LA) offer computational means of converting raw interaction

logs into interpretable, topic-level diagnostics, and a growing body of research indicates that machine learning can identify at-risk students earlier and more consistently than manual review [1], [2]. Most of that research, summarized in Section I-A, frames the task purely as score prediction from quiz or clickstream data. A comprehensive evaluation of a student, however, goes beyond performance on multiple-choice items; it also depends on whether the learner can put an idea into words aloud. This paper considers both aspects together, with a quiz-based knowledge gap detector and a voice-based communication assessment module that share one login and one dashboard.

The paper makes three contributions. The first is a topic-level knowledge gap detection pipeline that groups quiz responses by subject and topic, computes the accuracy of each topic and flags every topic whose accuracy lies below a configurable mastery threshold, instead of relying on one overall percentage. The second is an independent communication assessment module that transcribes a spoken answer through the Web Speech API and scores it by keyword coverage against a subject-specific keyword bank, which provides a second and qualitatively different signal of understanding. The third is a candid evaluation of the current prototype, comprising functional verification of the core routes, a statistical profile of a 100-student cohort data set and a held-out machine-learning experiment; its results argue for collecting richer features before a predictive model can be trusted, rather than presenting an idealized outcome that the available data cannot support.

Three research questions guide the study. RQ1 asks whether topic-level accuracy can reveal weaknesses that an overall quiz score hides. RQ2 asks whether a simple and transparent threshold rule can flag those weaknesses in a form that a student or teacher can easily act on. RQ3 asks whether keyword-based scoring of a spoken answer can serve as a useful and interpretable companion to quiz-based assessment, even without any deep semantic understanding of the response. Sections II to IV return to these questions directly.

The rest of the paper is organized as follows. Section I-A reviews related work in educational data mining, knowledge tracing and adaptive learning and identifies the gap that this

paper addresses. Section II presents the system architecture, the knowledge gap and communication assessment models, and the implementation, covering the technology stack, the data pipeline and the system modules. Section III reports the results, Section IV discusses them together with the threats to validity, and Section V concludes the paper and outlines future work.

A. Related Work

EDM concentrates on applying computational and statistical techniques to extract useful knowledge from educational data sets. Romero and Ventura established it as a systematic discipline spanning data collection, pre-processing, pattern mining and pedagogical interpretation [1], and their later survey describes how the field moved from offline rule mining to real-time learning analytics as digital platforms produced ever larger interaction logs [2]. Learning Analytics is less about the mining algorithm itself and more about measuring, reporting and visualizing learner data so that instructors can act on it, with dashboards and cohort-progress views as its typical output. Knowledge tracing lies alongside both fields and poses a narrower question: given a sequence of earlier attempts, how probable is it that a learner has mastered a particular skill at this moment? Bayesian Knowledge Tracing by Corbett and Anderson answers it with a hidden Markov model that revises a mastery probability after each attempt [3], and Piech et al. later replaced its hand-specified transition structure with a recurrent neural network in Deep Knowledge Tracing, allowing the model to learn non-linear temporal patterns directly from attempt sequences [4]. Both approaches presume a continuous stream of attempts for each skill, a stronger data requirement than the present system satisfies, but they indicate the natural next step once enough attempt history has accumulated through the collection pipeline of Section II-D.

Supervised classification and regression are the most widely used tools for predicting student outcomes from historical and interaction data. Decision trees yield interpretable, rule-based predictions; Support Vector Machines, introduced by Cortes and Vapnik, cope well with high-dimensional behavioral feature spaces [6]; and ensemble methods such as Random Forest [5] and gradient boosting frameworks such as XGBoost [7] usually deliver the highest raw accuracy by averaging over many weak learners. The scikit-learn library used by this project's prediction service implements most of these models behind one common interface [8], which made Random Forest a convenient first baseline (Section III-C), although the results show that the single feature available to it is insufficient on its own.

Within this wider toolbox, one line of work predicts academic performance directly from assessment and demographic records. Bunkar et al. applied decision tree algorithms such as ID3 and C4.5 to classify pass or fail outcomes from marks and demographic indicators [9], and Thai-Nghe et al. used Bayesian Networks on the same kind of structured institutional record to estimate the probability of

success [10]. Koprinska et al. went beyond static records by passing Learning Management System clickstreams, submission timestamps and forum activity to a Support Vector Machine, and found that this dynamic interaction data was a stronger early-warning signal than demographic features alone [11]. For the larger feature sets that behavioral data creates, Yağcı compared Random Forest, Logistic Regression and Gradient Boosting for midterm-to-final grade prediction and found that tree-based ensembles generalized best across cohorts [12], while Fernandes et al. combined socio-economic indicators with enrollment history in a Random Forest model to study patterns associated with attrition [13]. In their review of this literature, Shahiri and Husain concluded that Random Forest and K-Nearest Neighbors reliably achieve high accuracy, but that the manual feature engineering each new data set demands limits their practical use [14]. More recent studies turn to deep temporal architectures that consume raw interaction sequences: Waheed et al. trained deep neural networks on LMS clickstreams and reported earlier identification of at-risk students than baseline classifiers [15], and Alturki et al. used LSTM and recurrent architectures to map chronological event streams, such as intervals between quiz attempts and video consumption, onto continuous performance trajectories [16]. Because such deep models are hard to interpret, Alamri and Alharbi combined them with the explainability tools SHAP [18] and LIME [19], so that a risk prediction can be traced back to the input features behind it [17]; this addresses the same concern that motivates the topic-level accuracy breakdown of Section II-B, although by a very different route.

Taken together, these studies reveal a clear and recurring trade-off. Static rule-based classifiers [9], [10] are easy to interpret but cannot represent how a student's understanding changes over time. Ensemble methods [12]–[14] are more accurate but are usually trained and deployed offline as a batch process, detached from the moment a student is working on a topic. Deep temporal models [11], [15], [16] close part of that gap by reading interaction sequences directly, yet their internal decision process remains opaque unless a separate explainability layer is added [17]. What almost none of this work does is act on its own prediction: a risk score is computed and a dashboard is refreshed, but a human instructor still decides what the student should practice next. Agentic and multi-agent AI is a newer direction that aims to close exactly this loop, letting a system both diagnose a learner's state and decide how to respond in real time by coordinating language-model-based agents with reinforcement-learning policies for personalization and remediation [20]. Table I positions the present system against this background. It sits deliberately nearer the interpretable end of the trade-off, reporting a per-topic score and a plain-language reason for each flag instead of a black-box risk probability, and, unlike the agentic systems in [20], it does not yet close the loop autonomously, so a student or teacher still decides what to do with a detected gap.

TABLE I
QUALITATIVE COMPARISON OF REVIEWED APPROACHES

Approach	Data used	Interpretable	Acts on result
Decision trees, Bayesian networks [9], [10]	Static marks and demographic records	High (explicit rules)	No
SVM with interaction logs [11]	Clickstreams, submissions, forum activity	Low	No
Ensembles: RF, GBM, KNN [12]–[14]	Grades, midterm marks, socio-economic data	Moderate	No (offline, batch)
Deep temporal models: DNN, LSTM, RNN [15], [16]	Clickstreams and event sequences	Low	No (risk score only)
Deep models with SHAP / LIME [17]	Same as [15], [16]	Moderate (post hoc)	Manual, by educators
Agentic / multi-agent AI [20]	Interaction data, natural language	Varies	Intended: autonomous
This work	Quiz accuracy per topic, voice keyword match	High (per-topic, score-based)	Suggests next topic to practice

Four structural limitations motivate this work despite such progress. First, most EDM platforms remain passive: they flag at-risk students on a dashboard without proposing a concrete next step. Second, standard pipelines depend on static, batch-processed snapshots instead of updating a learner profile as soon as a new quiz is submitted. Third, predictive models concentrate on structured numeric data and largely ignore unstructured signals such as a student's own spoken explanation of a concept. Fourth, the more accurate deep models tend to be the least interpretable, which makes it hard to give a student a plain reason why a topic has been flagged. This paper tackles the last two limitations directly at a modest and transparent scale: it adds a keyword-scored voice channel next to the quiz channel, and every flag the system raises traces back to one visible number, the accuracy of the topic against its threshold, rather than to the internal weights of a model.

II. METHODS

A. System Architecture

The system, built under the working name Atlas, follows the pipeline in Fig. 1. A student signs in with an institutional roll number, which the backend checks against a cohort record instead of a password, because the present prototype targets a single classroom rather than public access. After signing in, the student can attempt a subject quiz, and the chosen options are scored immediately against an answer key. The per-question outcomes are written to a raw CSV record through a data-collection endpoint that checks required fields and value ranges, and are later cleaned into a processed record holding

only the student, subject, topic and computed accuracy. Processed records are grouped by topic to drive the gap detection step of Section II-B, and the weakest flagged topic is shown as a recommendation. Separately from the quiz, the communication module (Section II-C) records a spoken answer to an open-ended prompt and scores it by keyword coverage. Quiz history and communication results are returned to the student on a dashboard with Overview, Quizzes, Video library, AI tutor, Skill map and History pages, so a flagged gap and a suggested next topic appear beside the content the student would use to work on it.

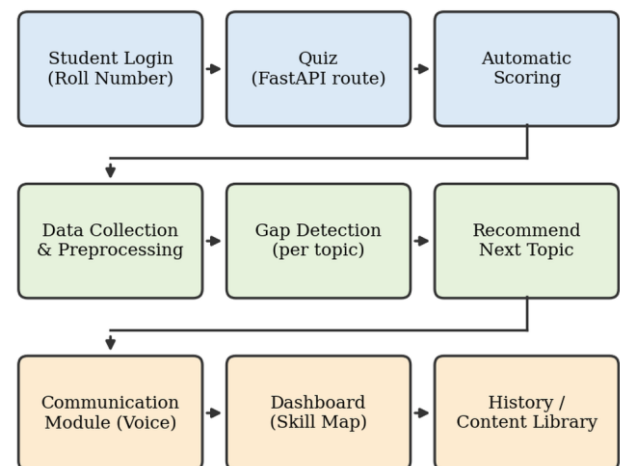


Fig. 1. Pipeline of the implemented system, from student login through quiz scoring, data preparation, topic-level gap detection, and the independent voice-based communication module.

B. Knowledge Gap Detection Model

Every quiz response is stored as a correct/total pair for a student, subject and topic. The accuracy of one topic attempt is

$$\text{Accuracy} = (\text{Correct Answers} / \text{Total Questions}) \times 100 \quad (1)$$

and the processed data set in Table II holds one such value for each student–subject–topic combination, once the raw pairs have been validated and normalized to this common scale. A topic is flagged as a knowledge gap when its accuracy drops below a configurable mastery threshold θ , currently 70%:

$$\text{Gap}(\text{topic}) = 1 \text{ if } \text{Accuracy}(\text{topic}) < \theta, \text{ else } 0 \quad (2)$$

Among the flagged topics, the system currently recommends the first one returned by the gap check as the next item for practice; since topics are evaluated independently, this equals recommending the weakest flagged topic whenever the list is produced in ascending order of accuracy. Table III gives a short worked example of this exact rule: four topics attempted by one student, of which only Data Structures falls under the 70% threshold and is therefore the topic recommended for the next practice session.

TABLE II
PROCESSED ASSESSMENT RECORD FIELDS

Field	Description
student_id	Roll number linking the record to a learner profile
subject	Academic subject (e.g., Mathematics, Physics)
topic	Sub-topic within the subject (e.g., Algebra, Mechanics)
accuracy	Percentage score for that topic, computed via Eq. (1)

TABLE III
WORKED EXAMPLE OF TOPIC-LEVEL GAP DETECTION

Topic	Accuracy	Gap($\theta=70$)	Rec.
Algebra	90%	No	
Mechanics	70%	No	
Data Structures	60%	Yes	Next
Writing	80%	No	

Algorithm 1 summarizes the complete procedure.

ALGORITHM 1
TOPIC-LEVEL GAP DETECTION AND RECOMMENDATION

Procedure for one student
Input: processed records of the student; mastery threshold θ (70 by default).
1. Group the records by subject and topic.
2. For each topic, compute Accuracy = (Correct Answers / Total Questions) $\times$ 100.
3. If Accuracy $< \theta$, add the topic to the gap list.
4. Take the first topic returned by the gap check as the recommendation; this is the weakest flagged topic when the list is in ascending order of accuracy.
Output: the gap list and the recommended topic.

The rule is intentionally simple. It is a single threshold on a directly observable number, so a student or teacher can see precisely why a topic was flagged without interpreting any model weights. As noted in Section I-A, the wider EDM literature often formulates a richer weighted composite that also folds in response latency, question difficulty and a student's historical trend on the same topic, each normalized to $[0, 1]$ and combined as

$$G = w_1(1 - A_n) + w_2T_n + w_3P_n + w_4D_n \quad (3)$$

where A_n , T_n , P_n and D_n denote the normalized accuracy, response time, prior weakness and difficulty of a topic, and the w_i are non-negative weights that sum to one. The present data schema (Table II) does not yet record response time or a difficulty rating for each question, so Eq. (3) is not computed in the running system; it is kept here as the direction in which the gap model is meant to grow once that telemetry is gathered through the same data-collection endpoint, and not as a claim about the current implementation.

C. Communication Assessment Module

The communication module assesses a student's spoken explanation independently of the quiz. A subject-specific open-ended prompt is displayed along with a bank of expected keywords for it; the student's answer is captured through the browser's Web Speech API, transcribed to text in real time and compared with the keyword bank. The communication score is

$$\text{Score} = (\text{Matched Keywords} / \text{Expected Keywords}) \times 100 \quad (4)$$

and the result falls into one of three feedback tiers: "Needs review" below 50%, "Good effort" from 50% up to 80%, and "Strong answer" at 80% or above. As an example, for the Physics prompt "Explain Newton's second law in your own words", whose keyword bank is {force, mass, acceleration, newton, second, law, motion}, the transcript "Newton's second law says force equals mass times acceleration" matches six of the seven keywords, missing only motion, which gives 86% and the "Strong answer" tier. Keywords absent from the response are also listed back to the student, so a lower score comes with a concrete suggestion of what to add instead of a bare number. The keyword bank is defined for each subject, and the present catalog covers twenty-two subjects, from Mathematics and Physics to Economics, Psychology and Media Literacy, so the module is not tied to the content of one department.

Table IV lists the three feedback tiers used for the communication score.

TABLE IV
COMMUNICATION FEEDBACK TIERS

Score range	Feedback tier
Below 50%	Needs review
50% up to 80%	Good effort
80% or above	Strong answer

D. Technology Stack and Data Pipeline

The backend is written in Python with FastAPI, and Pydantic models validate every request and response body. It is split into separate route and service modules, so that, for instance, the gap detection rule of Section II-B is one small, independently testable function and not logic buried in a route handler. The frontend is a static dashboard in plain HTML, CSS and JavaScript that calls the backend through its JSON API, so the client needs no build step. The prediction service uses scikit-learn's RandomForestRegressor, saved with joblib so that a trained model can be reloaded without retraining on each request. Table V lists the main API endpoints. Data currently persists as CSV files under version control; the project's own documentation already describes this as a prototype-scale choice and names SQLite for a single-instance deployment or PostgreSQL for a shared multi-worker deployment as the migration path once concurrent writes outgrow what a lock around a single CSV file can handle safely.

TABLE V
MAIN API ENDPOINTS

Method	Endpoint	Purpose
GET	/api/students/{id}	Resolve a roll number to a learner profile
GET	/api/quizzes/recent	List a student's recent quiz results
POST	/api/quizzes/submit	Score a submitted quiz against the answer key
POST	/api/data/collect	Validate and append raw performance records
POST	/api/data/preprocess	Clean raw records into per-topic accuracy
GET	/api/performance/summary	Return cohort-level mastery and accuracy
GET	/api/content	List the learning-content catalog by subject
GET	/api/assessments/{topic}	Return a topic assessment, answers hidden
POST	/api/assessments/{topic}/submit	Grade an assessment and return feedback

E. System Modules

The dashboard is organized into six primary pages: Overview, which shows progress and analytics at a glance; Quizzes, for content delivery and assessment; a Video library of structured learning content; an AI tutor page for personalized practice; a Skill map that shows knowledge gaps and related topics to practice next; and History, which lists collected activity and processed results. A Profile page is available as a secondary account page. The learning-content catalog currently contains twenty-two entries across subjects that include Mathematics, Physics, Computer Science, English, Biology, Chemistry, History, Economics, Psychology and Media Literacy, and each entry carries a content type, a duration and a difficulty level so that a flagged topic can be linked straight to a relevant video or lesson. The assessment bank at present ships two fully worked topics, quadratic equations and mechanics, each with multiple-choice questions, a correct option and a short explanation that is returned only after grading, so the answer key never reaches the client before submission. For each of the other twenty subjects in the catalog, the quiz module generates a short study-habit-oriented question set, so every subject in the content library has a quiz to accompany it even before a dedicated question bank has been written.

F. Evaluation Protocol and Metrics

The evaluation combines three complementary checks. Functional verification confirms that each core service returns the expected output for representative inputs. The cohort analysis characterizes the mastery distribution on which the threshold acts. The predictive experiment measures how well a baseline model estimates topic accuracy for students it has not seen. For the last check, prediction quality is reported with two standard measures. The mean absolute error over n held-out

records is

$$MAE = (1/n) \sum |y_i - \hat{y}_i| \quad (5)$$

where y_i and $\hat{y}_i$ are the actual and predicted accuracy of the i -th record, and the coefficient of determination is

$$R^2 = 1 - \Sigma(y_i - \hat{y}_i)^2 / \Sigma(y_i - \bar{y})^2 \quad (6)$$

with $\bar{y}$ the mean of the actual values. A negative R^2 means that the model fits the held-out data worse than a constant prediction equal to their mean, which is why Section III-C reports a mean-prediction baseline next to the model.

III. RESULTS

A. Functional Verification

The core service functions were run directly on representative inputs, instead of only being inspected in the code. Table VI reports the observed output of each case. The gap detection and recommendation check used the worked example of Table III, and the grading checks used the genuine three-question quadratic-equations bank, once with every answer correct and once with a single wrong answer, to confirm that partial credit and feedback are both computed correctly.

TABLE VI
FUNCTIONAL VERIFICATION, WITH OBSERVED OUTPUT

Case	Input	Observed output
Gap detection	{Algebra:90, Mechanics:70, Data Structures:60, Writing:80}, $\theta=70$	Gaps = [Data Structures]
Topic recommendation	Gaps = [Data Structures]	Recommended = Data Structures
Assessment grading, all correct	quadratic-equations, answers [1,2,2]	Score 3/3 (100%), no feedback items
Assessment grading, one wrong	quadratic-equations, answers [1,0,2]	Score 2/3 (67%), 1 feedback item returned
Communication scoring	Physics transcript (Section II-C)	6/7 keywords matched, 86%, Strong answer

B. Cohort Data Analysis

The project's seed cohort of 100 students was analyzed to characterize the mastery distribution on which the gap detection threshold acts. The mean mastery of the cohort is 83.4%, with a minimum of 67.0% and a maximum of 97.0%. Fig. 2 shows the full distribution against the 70% gap threshold of Section II-B. Eight students, roughly one in twelve, fall below that threshold on their overall mastery figure, and nineteen fall below a more conservative 75% line. This illustrates why a single overall number is a coarse signal: a student who sits just above the 70% line overall can still be far below it on one specific topic, which is precisely the case that topic-level accuracy is designed to detect. Table VII collects the cohort figures.

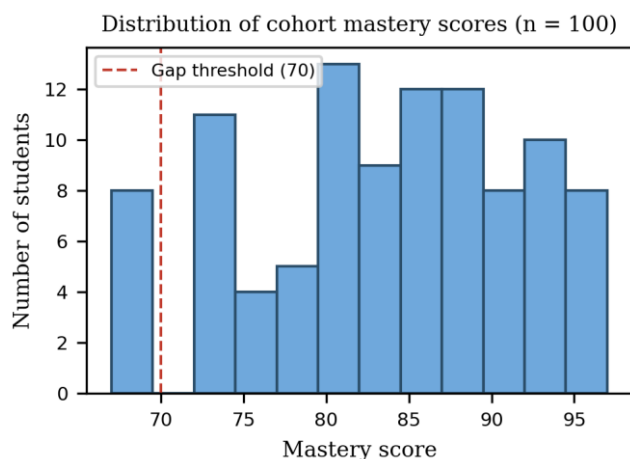


Fig. 2. Distribution of overall mastery across the 100-student cohort data set, with the 70% gap-detection threshold marked.

TABLE VII
COHORT MASTERY SUMMARY

Measure	Value
Students in the cohort	100
Mean mastery	83.4%
Minimum mastery	67.0%
Maximum mastery	97.0%
Students below the 70% threshold	8
Students below the 75% line	19

C. Predictive-Modelling Experiment

To test the RandomForestRegressor baseline described in Section II-D, a data set of 160 records was generated through the system's own collection and preprocessing endpoints: 40 students, each with one quiz attempt in each of four subjects, with correct/total pairs sampled so that a student's results loosely follow their cohort mastery plus random noise. The 40 students were split by student ID into a training set of 30 and a test set of 10, so that no student's records appear in both halves, and a RandomForestRegressor was trained with student ID as its only input feature, exactly as the project's current training script does. On the held-out students this model reached a mean absolute error of 15.3 accuracy points and a negative R^2 of -0.74 , which means it did worse than simply predicting the training-set mean accuracy for every student; that trivial predictor had a mean absolute error of 11.4 points. Fig. 3 plots predicted against actual accuracy for a sample of the held-out students, and Table VIII summarizes the set-up and outcome of the experiment.

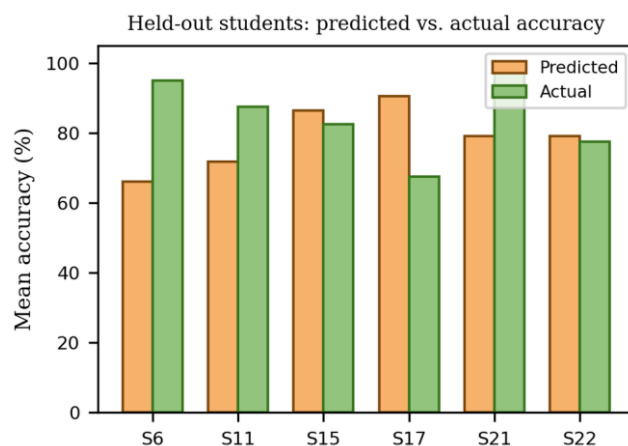


Fig. 3. Predicted vs. actual mean accuracy for six held-out students, using student ID as the model's only feature.

TABLE VIII
SET-UP AND OUTCOME OF THE PREDICTIVE EXPERIMENT

Item	Detail
Data set	160 records, 40 students, four subjects
Split by student ID	30 training, 10 test
Input feature	Student ID only
Mean absolute error	15.3 accuracy points
R^2	-0.74
Mean-prediction baseline error	11.4 accuracy points

IV. DISCUSSION

A. Interpretation of the Findings

The predictive result is informative rather than disappointing. A Random Forest trained only on student ID cannot generalize to an ID it has never seen, because the model can only interpolate between the average outcomes of the training IDs, so its failure on unseen students is expected and is not a software defect. The experiment confirms by measurement, and not by assumption, that student ID alone is not a usable predictive feature, and that the accuracy, response-time, difficulty and prior-performance features named in Eq. (3) are necessary, not optional, when the system moves from topic-level thresholding toward predictive gap scoring.

Returning to the research questions, the cohort analysis (Section III-B) shows that topic-level accuracy exposes weaknesses that an overall figure hides (RQ1), and the threshold rule (Section II-B) allows every flag to be traced to one visible number (RQ2). The worked example in Section II-C shows keyword coverage to be a usable and interpretable signal even without semantic understanding (RQ3), although it checks the presence of terms and not the correctness of the reasoning.

Table IX lists, for each research question, the evidence from the preceding sections that bears on it.

TABLE IX
RESEARCH QUESTIONS AND SUPPORTING EVIDENCE

Question	Evidence	Finding
RQ1	Cohort analysis (Section III-B)	Topic-level accuracy exposes gaps that the overall figure hides
RQ2	Threshold rule (Section II-B)	Every flag traces to one visible number
RQ3	Communication example (Section II-C)	Keyword coverage is usable and interpretable, but checks term presence only

B. Limitations and Planned Remedies

Several constraints of the present prototype follow directly from the design choices described earlier, and each has a remedy that the design already anticipates. The roll-number login and the CSV storage suit a single classroom but not a shared deployment. The regression baseline uses student ID alone because the data schema does not yet record response time or question difficulty. Keyword matching rewards the presence of terms and not the correctness of an explanation. The threshold rule, although transparent, judges each topic from its recorded accuracy without maintaining a continuously updated estimate of mastery, and the system flags a gap without deciding how to close it. Table X pairs each of these limitations with the remedy proposed in this paper.

TABLE X
CURRENT LIMITATIONS AND PLANNED REMEDIES

Limitation	Planned remedy
Single-file CSV storage	Move to SQLite or PostgreSQL
Student ID as the only model feature	Collect response-time, difficulty and prior-performance features
Keyword matching checks term presence	Add sentence-embedding similarity scoring
Static threshold on recorded accuracy	Knowledge-tracing mastery estimate once attempt history exists
Gap flagged but response left to the user	Agentic recommendation layer

C. Threats to Validity

1) *Internal Validity*: The regression experiment in Section III-C predicts the same accuracy metric that the gap detection rule in Section II-B already computes directly from the same records, so any future model should be validated against an independent label, such as a later re-assessment of the same topic, and not against a value derived from its own training features.

2) *External Validity*: Both the 100-student cohort data (Section III-B) and the 160-record predictive-modelling data set (Section III-C) are seed or simulated data and not data collected from real classroom use, so the absolute figures reported here should not be read as a claim about the performance of a deployed system; the same pipeline, re-run on real collected data, is expected to behave differently.

3) *Construct Validity*: Keyword matching in the communication module only tests whether the expected terms are present, not whether the sentence around them explains the concept correctly, so a transcript that lists keywords without connecting them meaningfully can still score well; the conclusion acknowledges this as a target for semantic and not lexical evaluation.

V. CONCLUSION

This paper presented a knowledge gap detection system that groups quiz responses by topic instead of relying on one overall score, flags every topic that lies below a mastery threshold and recommends the weakest flagged topic for further practice, together with an independent voice-based communication module that scores a spoken answer by keyword coverage against a subject-specific bank. Looking back at the research questions, the cohort analysis showed that topic-level accuracy exposes weaknesses that an overall mastery figure can conceal; the threshold rule is simple enough that every flag traces back to one visible number; and the worked communication example showed that keyword coverage gives a usable, interpretable signal even without full semantic understanding of the response, although this signal checks for the presence of terms and not for correct reasoning.

The predictive-modelling experiment points most clearly to what should happen next. Before any model is used to predict or anticipate a knowledge gap, the system needs to collect the response-time, difficulty and historical-performance features of the extended gap score, because student identity alone was shown not to generalize to new students. Three extensions follow naturally from the current design. First, the keyword matching of the communication module can be replaced or supplemented by semantic similarity scoring based on sentence embeddings, so that a correct explanation phrased in different words is rewarded. Second, once enough per-topic attempt history accumulates through the existing data-collection endpoint, a knowledge-tracing model such as Bayesian or Deep Knowledge Tracing could replace the static threshold with a continuously updated mastery estimate. Third, the CSV-based storage should move to SQLite or PostgreSQL, as the project's own documentation already recommends, both to support concurrent writes from many users and to make the richer feature set practical to query at scale. In the longer term, pairing a more capable gap and mastery model with an agentic recommendation layer could let the system move from flagging a gap to actively sequencing the practice that closes it.

ACKNOWLEDGMENT

The authors thank the Department of Artificial Intelligence and Data Science, Shridevi Institute of Engineering and Technology, Tumakuru, Karnataka, India, for providing academic guidance and institutional support throughout the development of this project. Generative AI (Claude, developed by Anthropic) was used to assist with formatting the manuscript to the journal template and with language editing; the authors

reviewed all content and take full responsibility for it.

REFERENCES

- [1] C. Romero and S. Ventura, *Educational data mining: A review of the state of the art*, IEEE Trans. Syst., Man, Cybern. C, vol. 40, no. 6, pp. 601–618, 2010.
- [2] C. Romero and S. Ventura, *Educational data mining and learning analytics: An updated survey*, WIREs Data Mining Knowl. Discov., vol. 10, no. 3, 2020.
- [3] A. T. Corbett and J. R. Anderson, *Knowledge tracing: Modeling the acquisition of procedural knowledge*, User Model. User-Adapt. Interact., vol. 4, pp. 253–278, 1995.
- [4] C. Piech et al., *Deep knowledge tracing*, in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 28, 2015, pp. 505–513.
- [5] L. Breiman, *Random forests*, Mach. Learn., vol. 45, pp. 5–32, 2001.
- [6] C. Cortes and V. Vapnik, *Support-vector networks*, Mach. Learn., vol. 20, pp. 273–297, 1995.
- [7] T. Chen and C. Guestrin, *XGBoost: A scalable tree boosting system*, in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 785–794.
- [8] F. Pedregosa et al., *Scikit-learn: Machine learning in Python*, J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
- [9] K. Bunkar, U. K. Singh, B. Pandya, and R. Bunkar, *Data mining: Prediction for performance improvement of graduate students using classification*, in Proc. Int. Conf. Wireless Opt. Commun. Netw. (WOCN), 2012.
- [10] T. Thai-Nghe et al., *Academic performance prediction using machine learning techniques*, in Proc. IEEE Int. Conf. Intell. Syst. Design Appl. (ISDA), 2011, pp. 412–417.
- [11] I. Koprinska et al., *Predicting student performance in online learning environments using clickstream logs*, IEEE Trans. Learn. Technol., vol. 8, no. 4, pp. 354–365, 2015.
- [12] M. Yağcı, *Educational data mining: Prediction of students' academic performance using machine learning algorithms*, Smart Learn. Environ., vol. 9, no. 1, Art. no. 11, 2022.
- [13] E. Fernandes et al., *Educational data mining to predict student academic performance in public schools*, Comput. Educ., vol. 136, pp. 115–129, 2019.
- [14] A. M. Shahiri and W. Husain, *A review on predicting student's performance using data mining techniques*, Procedia Comput. Sci., vol. 72, pp. 414–422, 2015.
- [15] H. Waheed, S.-U. Hassan, N. R. Aljohani, J. Hardman, S. Alelyani, and R. Nawaz, *Predicting academic performance of students from VLE big data using deep learning models*, Comput. Hum. Behav., vol. 104, Art. no. 106189, 2020.
- [16] S. Alturki et al., *Temporal student performance prediction using Recurrent Neural Networks and LSTM*, IEEE Access, vol. 9, pp. 128450–128462, 2021.
- [17] A. Alamri and S. Alharbi, *Explainable AI for student performance prediction in digital learning platforms*, IEEE Trans. Educ., vol. 66, no. 3, pp. 280–289, 2023.
- [18] S. M. Lundberg and S.-I. Lee, *A unified approach to interpreting model predictions*, in Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 4765–4774.
- [19] M. T. Ribeiro, S. Singh, and C. Guestrin, *'Why should I trust you?': Explaining the predictions of any classifier*, in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 1135–1144.
- [20] S. Hosseini and H. Seilani, *The role of agentic AI in shaping a smart future: A systematic review*, Array, vol. 26, Art. no. 100399, 2025.