



A Neuro-Symbolic Feedback Engine for Evidence-Grounded Feedback to At-Risk Students, with an LLM-as-a-Judge

Swathieswari Mohanraj¹, Shanmugavadivu P²

¹Research Scholar, Department of Computer science and applications, Gandhigram Rural Institute, Gandhigram, Dindigul 624302, Tamil Nadu, India.

²Senior Professor, Department of Computer science and applications, Gandhigram Rural Institute, Gandhigram, Dindigul 624302, Tamil Nadu, India.

swathieswari.mca@gmail.com¹, psvadivu@ruraluniv.ac.in²

Article history

Received: 10 August 2026

Accepted: 07 September 2026

Published: 30 September 2026

Keywords:

Early warning; Neuro-Symbolic AI; Explainable AI; surrogate decision tree; large language model feedback; LLM-as-a-Judge; Learning Analytics.

Abstract

Early-warning models can flag students at risk, but they do not say why a student is at risk or what feedback should follow. This study examines a neuro-symbolic pipeline in which a trained model makes the prediction, a symbolic rule layer explains it, and the evidence behind the prediction is checked against the student's data before feedback is issued. Seven models such as Logistic Regression (LR), Random Forest (RF), Extreme Gradient Boosting (XGBoost), Multilayer Perceptron (MLP), and Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM) and Gated Recurrent Unit (GRU) were trained on 2013 students from the Open University Learning Analytics Dataset and tested on 2014 students, to predict four outcomes (Distinction, Pass, Fail and Withdrawn) for 10,740 students from records up to the midpoint of each module. All models beat the majority-class baseline (Macro-F1 0.455 to 0.508 against 0.170). Random Forest scored the highest, while the MLP was more sensitive to students needing support, identifying 35% of Withdrawn students against 15%. Shallow decision trees fitted to each model's predictions gave readable if-then rules that matched the random forest for 82% of students and the MLP for 72%, compared with 49% and 36% for the majority class. The rules held steady under resampling and carried over from 2013 to 2014 students' data, and its features influenced the models about three times as much as random features. A language model then generated feedback for 100 students per model, and each claim was checked against the student's data. Feedback generated from the students' values alone contained incorrect claims in 10% (MLP) and 7% (random forest) of messages and supplying the model's most influential features together with the claim check brought this to 0% for both. The rule alone did not help (28% and 25% incorrect claims), so the gain came from the model's evidence and the check, with the rules making the model's behaviour readable and testable. The neuro-symbolic combination therefore produced feedback with fewer false claims. Thus, the trained model paired with rules and claim check provides reliable early warning feedback for students.

1. Introduction

Learning platforms record detailed traces of student activity, and its research has used such records to identify students at risk of failure or withdrawal, while timely intervention remains possible [1] – [3]. The Open University Learning Analytics Dataset (OULAD), which links student demographics, assessment outcomes and daily activity in a virtual learning environment [4], was often used benchmark for these tasks. Studies based on it indicate that clickstream data, assessment data, or a combination of the two can signal risk at an early stage [5] – [8]. Identifying an at-risk student is only the primary step. Educators additionally require an account of why a student is at risk and guidance on how to respond. Two tasks therefore follow prediction: explaining the model and converting the explanation into feedback that a student can act on. XAI methods such as SHAP and LIME, are the standard means of interpreting a model [9], [10]. The outputs of these methods are not always consistent, since different explainers may rank features differently for the same model and the same dataset [11]. An alternative is to induce a compact set of if-then rules from the predictions of the model itself [12], [13]. Such rules are readable and verifiable, but such rules constitute only an approximation of the model. An approximation that is inaccurate, unstable under resampling, or non-transferable to a new cohort provides no valid explanation. The evaluation of interpretability requires rigorous testing [14], yet none of the studies reviewed in Section II applies such testing to a rule layer. Large language models (LLMs) can generate written feedback from explanatory information [15] – [17]. Though, LLMs may produce factually incorrect statements, and erroneous feedback can mislead learners [18], [19]. One way to reduce this risk is to give the language model the evidence behind a prediction and then check what it says against the student's data [19]. A framework that passes SHAP output to a language model has been proposed for decision support [20], but none of the studies reviewed compares feedback written with and without the model's evidence while checking the resulting claims against student data. Two gaps remain. In the reviewed literature, a symbolic rule layer has not been examined for how closely it follows the model, how stable its rules are,

or whether the rules hold on a new cohort [12], [13],

and feedback from language models has not been tied to the model's evidence and checked against it [15] – [17]. This study therefore asks whether a symbolic rule layer can draw meaningful, evidence-based information about students from a neural model, and whether that information supports reliable feedback.

1.1. Research questions.

- RQ1: How well do neural networks and machine-learning models predict student outcomes early in a module, on a new cohort?
- RQ2: How accurate, stable and transferable is a symbolic rule layer fitted to a model's predictions?
- RQ3: Does feedback provided by model's evidence and checked against students' information contain fewer false claims?

This paper makes four contributions. First, seven models are compared on OULAD [4] using a temporal split, with training in 2013 students' data and testing in 2014 students' data. Second, a symbolic rule layer is evaluated on five criteria: fidelity, stability, transferability, agreement with SHAP, and the dependence of the model on the features named in the rules. Third, five versions of LLM-generated feedback are compared using a claim check against student data; the agreement of LLM-as-a-Judge [21] with that check is measured, and an ablation separates the contributions of the evidence, the rule and the check. Fourth, the limits of the findings are stated explicitly: the neural model is not shown to outperform the random forest, and the rule layer alone is not shown to reduce false claims.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the data, models, symbolic layer, feedback generation and checks. Section 4 reports the results, Section 5 discusses the results and the limitations of the study, and Section 6 concludes with the evidences.

2. Related Work

The literature on early warning, knowledge tracing, explainability, neuro-symbolic modelling, LLM-generated feedback and fairness was searched

systematically. Table 1 summarises the six most relevant studies of the last five years (2021–2026), and all reviewed studies are discussed thematically

below. Table 1 Relevant studies of the last five years (2021–2026)

Table 1 Relevant studies of the last five years (2021–2026)

Reference	Year	Topic	Method and data	Main finding	Limitation
[8]	2024	Early warning	Multidimensional time series; OULAD	Behaviour and assessment histories predict performance	Early-risk result uses a different task
[11]	2022	Explanation	Several explainers compared; MOOCs	Explainers disagree on the same model and students	Effect on feedback not tested
[44]	2023	Neuro-fuzzy	Neuro-fuzzy model with rules; OULAD clicks	Outcomes explained by fuzzy rules	Few comparisons and no external cohort
[13]	2024	Neuro-symbolic	Neural network with symbolic educational knowledge	More trustworthy, interpretable prediction	Tested only on computational-thinking data, with no risk prediction or feedback.
[19]	2026	LLM feedback	Systematic review of empirical studies	Hallucination and over-positive feedback recur	Evidence is mostly short-term and judged by experts, not by long-term learning outcomes
[20]	2026	LLM feedback	SHAP, case retrieval and GPT narratives; attrition	Explanations grounded in SHAP and cases	No learning or cohort-stability test

2.1. Early warning of at-risk student

The prediction of failure and withdrawal from online activity is a well-established research area. An early study on massive open online courses (MOOCs) produced weekly predictions for two offerings of one course and applied probability smoothing and transfer learning to stabilize the predictions [1]. In a university course of 76 students, unsuccessful students were found by week 3 [2], although a cohort of that size limits the generalisability of the result. A variety of model families have been applied to OULAD. A long short-term memory network trained on clickstream data achieved high precision but lower recall for the pass and fail classes [5], which illustrates that a single aggregate metric can conceal missed students. Other studies compared ensemble, neural and regression models [6], applied deep learning to

early activity data [22], modelled assessment trajectories with hidden Markov models [7], and analysed multidimensional time series [8]. Combining weekly activity with assessment information outperformed activity alone [23], and predictive performance improved as the course progressed [3]. A later cutoff thus provides more evidence but leaves less time for intervention. Further work employed temporal feature engineering [24], [25], real-time daily prediction [26], and a single neural network across multiple courses [27]. The last of these predicts the timeliness of assignments rather than the final result; it therefore demonstrates the importance of context without validating an OULAD outcome model. Like these, the present study predict outcomes using seven models and further by testing

the explanation and the feedback followed by prediction

2.2. Knowledge tracing and interpretable student models

Knowledge tracing models the acquisition of skills over time. The original formulation estimated the probability that a student had mastered each production rule taught by a tutor [28]. Deep learning sequential models followed [29], succeeded by attention-based models over previous responses [30], [31] and models incorporating behavioural signals such as response time, attempts and hints [32]. A parallel line of work seeks interpretability through causal relations [33], cognitive models of programming [34], fuzzy reasoning [35] and knowledge ontologies [36]. Knowledge tracing has also been embedded in a metacognitive tutor [37]. These approaches predict the next response or skill state in structured tutoring tasks. The present study instead predicts a course outcome at a single cutoff and does not model skill acquisition; it is therefore not a knowledge tracing study.

2.3. Explainable AI in education

LIME [10] and SHAP [9] are the most widely used model-agnostic explainers. In education, these explainers have been applied to learning analytics in virtual learning environments [38], to individualized support for at-risk students through counterfactual explanations [39], to personalized intervention following dropout prediction [40], and within interpretable prediction pipelines [41], [42]. One study converted explanations into action recommendations to support self-regulation and reported improved learning outcomes [43]. Fuzzy rules have also been proposed to explain learning analytics on OULAD [44]. A plausible explanation is not necessarily a faithful one. Across five MOOCs, different explainers produced different feature rankings for the same model and students [11], and teachers' judgements of explanations have been examined separately [45]. The present study adds evaluation criteria specific to a rule layer: the fidelity of the rules to the model, the stability of the rules under resampling, and the transferability of the rules to a new cohort.

2.4. Symbolic and Neuro-symbolic approaches in education

Neuro-symbolic AI integrates data-driven learning with symbolic knowledge and has been characterised as a third wave of AI [46]. In

education, symbolic educational knowledge has been incorporated into deep networks to improve trustworthiness and interpretability [13], and a neuro-symbolic multi-modal architecture has been proposed for adaptive tutoring [47], although no evaluation is reported for the latter. Fuzzy and ontology-based models likewise provide readable structure [35], [36], [44]. In the present study, the symbolic layer is not authored by experts but induced by a decision tree from the predictions of a trained model [12], [48], and its quality is measured.

2.5. Language-model feedback and its reliability

LLMs have been used to generate formative feedback, including cognitive and metacognitive prompts [15] and feedback within the tutoring systems [17]. A recent review reports that hallucination, over-positive feedback and misjudgment of student work recur across studies, and that grounding and teacher oversight are possible safeguards [19]. Feedback design can draw on research in intelligent tutoring systems and the learning sciences [16], and generative AI has been applied to support self-regulated learning [49]. The effects of erroneous LLM feedback have been examined in mathematics tutoring [18]. The work closest to the present study is a framework that combines SHAP output, retrieved cases and an LLM to support decisions on student attrition [20]. The present study likewise places the model's evidence in the prompt, and adds a claim check against student data together with a comparison of versions with and without each component. LLMs are also used to judge the output of other models [21], and the literature documents biases in such judges, including a preference for text produced by the same model. Language-model judges are known to favour text from their own model [21], so a Judge is used here only as a second check, and its agreement with the exact claim check is measured.

Prediction models do not always serve every group of students equally well. A slicing analysis of 44 MOOCs found that fairness depended on the algorithm, the features, and the course [50]. On OULAD, removing protected attributes lowered measured fairness [51], and gender and disability differences appeared in early-warning decisions [52]. Fairness has also drifted over time in a national study of dropout prediction [53], and newer pipelines now report subgroup results next to

overall performance [42]. Most of the studies reviewed here treat prediction, explanation and feedback as separate problems. Work on early warning shows that outcomes can be predicted from student activity [5], [7], [8], work on explanation shows that explainers do not always agree [11], and work on feedback shows that generated text can be wrong [18], [19]. Among the reviewed studies, Among the studies reviewed, none reports a test of how closely a rule layer follows a model, how stable its rules are, or whether feedback tied to the model's evidence contains fewer incorrect claims. This study puts the three problems into a single pipeline and tests how they connect. how closely a rule layer follows a model, how stable its rules are, or whether feedback tied to the model's evidence contains fewer false claims. This study brings the three together in one pipeline and tests the links between them. Early-warning research shows that outcomes can be predicted from activity [5], [7], [8], explainability research shows that explanations can disagree [11], and research on feedback shows that generated text can be incorrect [18], [19]. This present study integrates prediction, explanation and feedback in a single pipeline and evaluates the links

among these components.

3. Methodology

This proposed methodology begins with early student records and ends with feedback that has been checked against the student's data (Fig. 1). Records up to the halfway point of each module are used to train seven models on the 2013 students, and the models are tested on the 2014 students. A small decision tree is then fitted to the predictions of a trained model, and the if-then rules read from this tree are passed to a language model together with the predicted outcome and the three most influential SHAP features. The language model writes the feedback, and every claim in the message is compared with the student's own data before the message is accepted. The design rests on the idea that the three parts complement one another: the neural or machine-learning model supplies the prediction, the symbolic layer turns the behaviour of the model into rules that can be read and tested, and the evidence together with the claim check keeps the generated feedback tied to what the model used for each student. Figure 1 shows Overview of the Proposed Neuro-Symbolic Feedback Pipeline

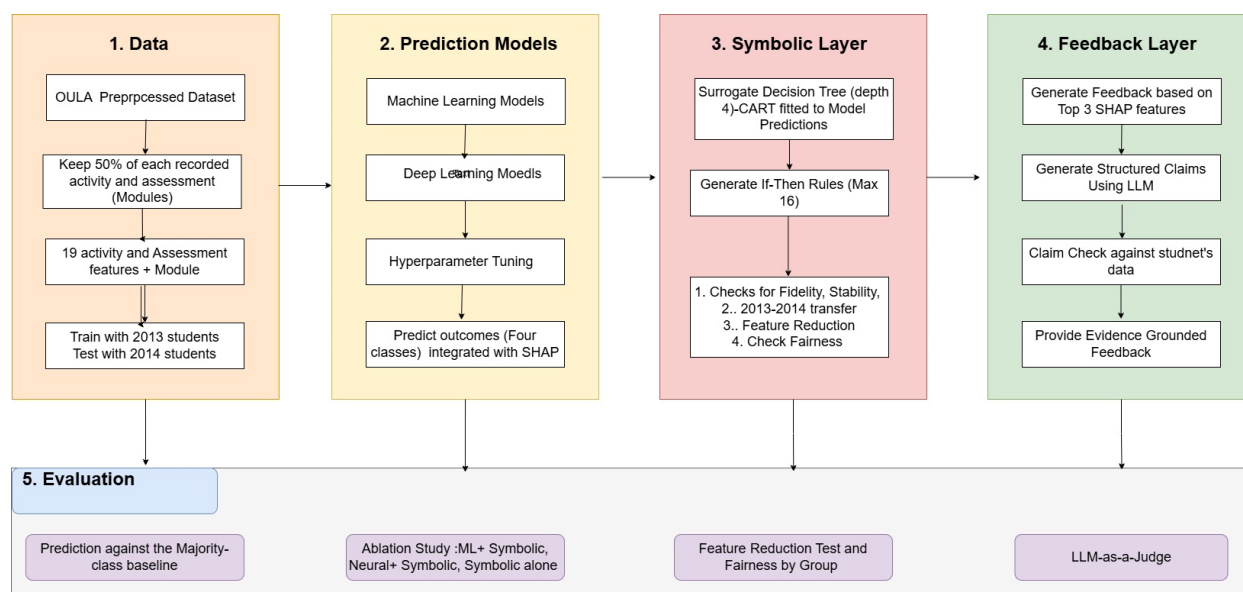


Figure 1 Overview of the Proposed Neuro-Symbolic Feedback Pipeline

3.1. Dataset

This research uses the Open University Learning Analytics Dataset (OULAD), a public and anonymised record of students on online courses at the Open University, United Kingdom. The dataset covers seven modules taught in 22 module presentations. A presentation is one run of a module

and starts in February (B) or October (J). OULAD holds 32,593 student enrolments from 28,785 students, 206 assessments with 173,912 submissions, 6,364 learning-platform materials, and about 10.6 million records of daily clicks. Four kinds of records were used in this proposed model.

Student background and result come from the student information table. Registration and withdrawal dates come from the registration table. Submission dates and scores come from the assessment tables. Daily clicks on learning materials come from the platform activity tables.

Student background fields (gender, age band, region, disability, education and deprivation band) are not model inputs and are used only in the fairness audit. Table 2 shows Dataset summary

Table 2 Dataset summary

Item	Value
Source	Open University Learning Analytics Dataset (OULAD)
Modules in OULAD	7 (6 used)
Module presentations in OULAD	22
Enrolments	32,593
Unique students	28,785
Assessments	206
Assessment submissions	173,912
Learning materials	6,364
Training set	2013 presentations, 10,643 students
Test set	2014 presentations, 10,740 students
Prediction time	50% of each module's length, in days
Target	Result: Distinction, Pass, Fail, Withdrawn
Majority class	Pass (51.6% in training, 51.7% in test)
Majority-class baseline	accuracy 0.517, macro-F1 0.170, balanced accuracy 0.250

Training and testing are separated by year. Presentations from 2013 form the training set and presentations from 2014 form the test set. This matches real use, where a model built by past students predicts for a new cohort. No 2014 data are used for training or tuning. Predictions are made at 50% of each module's length, and only data from before that point are used. Three groups of students

are removed (Table 3). The first is students who withdrawn before the cutoff, because such students are no longer at risk of the outcome. The second is test students in module CCC, which ran only in 2014 and has no training data. The third is test students who also appear in 2013, so that no student is in both sets. Six modules (AAA, BBB, DDD, EEE, FFF and GGG) remain.

Table 3 Students Excluded from The Training and Test Sets

Step	Training (2013)	Test (2014)
Enrolments	13,529	19,064
Withdrawal by the cutoff	-2,886	-5,106
Module CCC (no training data)	0	-2,864
2013	0	-354
Final	10,643	10,740

The target is the student's result, with four classes: Distinction, Pass, Fail and Withdrawn. The classes are not balanced (Table 4). Pass is the majority class in both years. A model that always predicts Pass reaches an accuracy of 0.517 but a macro-F1 of only 0.170. For this reason, macro-F1, which gives the four classes equal weight, is the main metric, and every model is compared with this baseline. Table 4 shows Class distribution

Table 4 Class distribution

Outcome	Training (2013)	Test (2014)
Pass (majority)	51.6%	51.7%
Fail	30.4%	26.7%
Distinction	10.1%	13.4%
Withdrawn	7.9%	8.2%

3.2. Features

Each student is described by 25 input features (Table 5): 19 behavioural and static features, and 6 module indicators, one for each module in the training data. All behavioural features use only information recorded up to the cutoff, which is 50% of the module length in days, rounded to a whole

day. Activity features come from the daily click records. These features describe how much a student has used the learning platform (total clicks, active days), how varied the use is (activity types, sites), when the student started (first active day), and whether the student is still active. The last of these is the share of clicks in the final third of the period before the cutoff. Assessment features cover non-exam assessments submitted by the cutoff. These features describe how many assessments were submitted, the scores, how late the submissions were, and how much of the assessed work the student has completed. Two flags mark students with no submission or no activity at all. Static features are the number of previous attempts at the module and the credits being studied. Counts are set to zero when a student has no records. Other missing values, such as the average score of a student with no submissions, are filled with the training median. Demographic fields are not model inputs. The sequence models (LSTM, BiLSTM and GRU) read the activity as 20 weekly steps. Table 5 shows Input features

Table 5 Input features

Group	Feature	Meaning
Activity	Total clicks	All clicks before the cutoff
Activity	Active days	Days with at least one click
Activity	Activity types	Number of different kinds of material used
Activity	Sites	Number of different materials opened
Activity	First active day	Day of the first click, counted from the module start
Activity	Recent clicks share	Share of clicks in the last third of the period
Activity	Clicks per active day	Total clicks divided by active days
Assessment	Submitted	Number of non-exam assessments submitted
Assessment	Mean score	Average score of submitted assessments
Assessment	Score spread	Standard deviation of the scores
Assessment	Mean lateness	Average days after the due date (negative if early)
Assessment	Late	Number of submissions after the due date
Assessment	Banked	Number of banked (carried over) submissions
Assessment	Weighted mastery	Weighted average score divided by 100
Assessment	Submission rate	Submitted divided by assessments due to the cutoff
Assessment	No submission	1 if the student submitted nothing, otherwise 0
Assessment	No activity	1 if the student had no clicks, otherwise 0
Static	Previous attempts	Earlier attempts at the module
Static	Studied credits	Credits the student is studying
Static	Module (6 indicators)	One indicator for each of AAA, BBB, DDD, EEE, FFF and GGG

3.3. Models and Tuning

Seven models predict the four outcome classes. There are three standard machine-learning baselines: logistic regression (LR), random forest (RF) [54] and XGBoost [55]. Four are neural networks: a multilayer perceptron (MLP) and three recurrent networks, the LSTM [56], BiLSTM [57] and GRU [58]. The MLP reads the 25 features in Table 5. The recurrent networks read the weekly activity of each student as 20 steps, together with the static and assessment features. The classes are unbalanced, so each model corrects for this during training. LR and RF use balanced class weights. XGBoost uses balanced sample weights. The MLP is trained on a sample in which the smaller classes are repeated until all classes are the same size. The recurrent networks use a class-weighted loss. The settings of each model were chosen on the 2013 data only. LR, RF, XGBoost and the MLP were tuned by random search with 10 trials, scored by macro-F1,

using five-fold cross-validation in which all records of one student stay in the same fold. The recurrent networks were tuned on a separate validation split of the 2013 data. Six settings were drawn at random from a grid of 12 (hidden size 32, 64 or 128; learning rate 0.001 or 0.003; dropout 0.2 or 0.4), and the best was kept. Training used the Adam optimizer [59], up to 40 epochs, a batch of 256, and early stopping after 5 epochs without improvement. The final recurrent models were trained with three random seeds, and the predicted probabilities were averaged. The 2014 test set was used once, after tuning, to report results. The main metric is macro-F1. Balanced accuracy, the area under the ROC curve (one class against the rest, averaged), and the recall of each class are also reported. Every model is compared with the majority-class baseline of Table 2.

Table 6 Models, Search Spaces and Chosen Settings

Model	Parameter Range	Chosen setting
Logistic regression	Regularisation strength C from 0.001 to 100	C = 21.4
Random forest	200 to 700 trees; depth none, 8, 12, 16 or 24; 1 to 9 students per leaf; features per split sqrt, log2 or half	388 trees, depth 16, 8 students per leaf
XGBoost	100 to 600 trees; depth 3 to 9; learning rate 0.01 to 0.3; row and column sampling; minimum child weight 1 to 9; L2 penalty 0.1 to 10	266 trees, depth 7, learning rate 0.079
MLP	Hidden layers (64), (128), (64, 32) or (128, 64); L2 penalty 0.00001 to 0.1; learning rate 0.0001 to 0.01	One hidden layer of 128 units, learning rate 0.00012
LSTM	Hidden size 32, 64 or 128; learning rate 0.001 or 0.003; dropout 0.2 or 0.4	128 units
BiLSTM	Same as LSTM	32 units
GRU	Same as LSTM	64 units

3.4. Symbolic Layer

The symbolic layer turns the predictions of a trained neural model into a small set of if-then rules that a teacher can read. The rules serve to describe the model, not to predict. The rules are a readable approximation of the behaviour of the model, and this research tests whether the approximation is accurate, stable and useful. A shallow decision tree (a surrogate) is trained to reproduce the model's

predictions, not the true outcomes. The tree has a maximum depth of four and at least 50 students in every leaf, and it uses the CART algorithm [48] with the Gini index to choose splits. For the tabular models, the training labels for the tree are the model's out-of-fold predictions on the 2013 students, so the model has not seen the student it is predicting. For the recurrent models, these labels are

the predictions of the final model on the training students, so the fidelity of these trees is not directly comparable with the other models. Each path from the top of the tree to a leaf becomes one rule: a set of conditions on the features (for example, mean score below a threshold and fewer than a given number of active days) and the predicted class. Five questions are asked of the rules (Table 7). The first is whether the tree reproduces the predictions of the model on students not used for training. The second

is whether the rules stay the same when the training sample changes. The third is whether rules learned from one cohort work on the next. The fourth is whether the rules point to the same features as SHAP [9], a standard method for explaining which features drove a prediction. The fifth is whether the features named in a rule are ones the model really depends on.

Table 7 Tests of the Symbolic Layer

Test	Question	How it is measured
Fidelity	Does the tree give the same answer as the model?	Share of 2014 test students for whom the tree and the model predict the same class, compared with always predicting the most common class and with a tree trained on the true results
Stability	Do the rules stay the same on a new sample?	30 bootstrap resamples of the 2013 students; the features used in the top three levels of each tree are compared by Jaccard overlap (shared features divided by all features used; 1 means identical)
Transfer	Do rules from 2013 work in 2014?	A tree learned in 2013, and a tree learned on half of the 2014 students are both scored on the other half of 2014
Agreement with SHAP	Do the rules and SHAP name have the same features?	Spearman rank correlation of feature importance, and overlap of the top five features
Deletion test	Does the model depend on the features in a rule?	For each student, the features named in the student's rule are replaced by the training median; the fall in the model's probability for its prediction and the share of predictions that change are compared with the same number of top SHAP features and of randomly chosen behavioural features

The deletion test uses 100 students. A paired Wilcoxon test [60] compares the rule features with the random features, and bootstrap intervals are reported for the differences. Fidelity is also reported for tree depths from 2 to 10, to show how much accuracy is lost by keeping the rules short. To separate the contribution of each part, three versions are reported: the neural or machine-learning model alone, the symbolic layer alone, and the model together with its rules. The comparison is made for the MLP and the random forest. The pipeline joins a predictive model, a symbolic layer and a verified feedback step, and each has a specific task that is

tested later. The model is there to predict, and a tree trained only on the true results is expected to do worse, which the ablation checks. The symbolic layer is there because a bare prediction cannot be inspected, so its rules are judged on fidelity, stability, transfer and the dependence of the model on the features named in the rules. The model's evidence and the claim check are included because a language model that sees only a student's values can say things the data do not support. Their worth is judged by the share of incorrect claims in the five feedback versions. The combination earns its place only if each part adds something measurable that the

others cannot.

3.5. Feedback Generation and Checking

This step asks whether feedback written by a language model stays true to the model's own evidence when that evidence is placed in the prompt and the output is checked. Feedback is written for 100 test students per model, chosen so that every predicted class is represented. GPT-4.1 writes all the messages at temperature 0. A single writer is used for every version, so that the versions differ only in the information given and not in who wrote them, which the ablation requires. GPT-4.1 was chosen because it follows instructions closely, which suits a fixed message structure and a judge that must answer in a fixed format. It is also available through an interface that fixes the model version and allows temperature 0, so the runs can be

repeated. It was a practical choice and was not selected by comparing models; the effect of the writer is listed as a limit. Each message follows a fixed structure: risk level, strengths, concerns, actions, a reflection question and a short message to the student. Depending on the version, the prompt contains some of the following: the student's own feature values with the median of each predicted class; the three behavioural features that influenced the prediction most, found with SHAP; and the rule that applies to the student. Five versions are compared (Table 8). The versions differ only in the evidence given and in whether the output is checked, so that each difference can be traced to one change. Table 8 shows Feedback Versions

Table 8 Feedback Versions

Version	Student values	SHAP features	Rule	Claim check
Ungrounded	Yes	No	No	No
SHAP only	Yes	Yes	No	No
Rule only	Yes	No	Yes	No
Grounded	Yes	Yes	Yes	No
Grounded and verified	Yes	Yes	Yes	Yes

A program checks every claim in a message against the student's real data. The rules of the check were fixed before the run. A message counts as incorrect if at least one claim names a feature that does not exist, states the incorrect direction compared with the class median (above or below), gives a value more than 5% from the true value, or gives the incorrect risk level. In the verified version, a failed message is regenerated once. If it fails again, a fixed template message replaces it. The tests check whether claims agree with the student's data and the model's evidence. It does not test whether a message agrees with the rule. A second automatic check uses a language model as-a-judge of the free text [21]. The judge is asked to act as a strict fact-checker. For each message it receives the evidence about the student (the predicted outcome and risk level, the feature values, the three most influential features with the SHAP values, and the rule that applies) and the feedback message. It lists every factual statement that the message makes about the student's data, behaviour, progress, trend or the

causes of the student's situation. Suggestions, encouragement and questions are ignored. It labels each statement as supported (follows from the evidence), unsupported (cannot be derived from the evidence, such as an invented trend, cause or past event) or contradicted (conflicts with the evidence), and returns the result in a fixed format. A message is counted as unsupported or contradicted if at least one of its statements has that label. Messages for which the judge's reply cannot be read are left out. The judge reviews the messages from three versions (ungrounded, grounded, and grounded and verified), 300 messages per model. It uses the same language model as the writer, so it can share the writer's blind spots, and it is treated as supporting evidence only. The judge is compared with the claim check using a cross-table and Cohen's Kappa [61]. Agreement between the judge and human readers was not measured.

3.6. Ablation Study

The ablation study removes or changes one part of the system at a time and measures what is lost. Each

Table 9 Design of the ablation study

Ablated component	What is compared	Measure
Learning model removed	A tree trained on the true results, against the trained models	Macro-F1
Rules fitted to the true results, not to the model's predictions	The two kinds of tree	Agreement with the model
Rule layer removed	The model with SHAP only, against the model with SHAP and rules	Prediction scores and measures of the rules
Tuning removed	Each model before and after tuning	Macro-F1, balanced accuracy, recall of Withdrawn
Weekly sequence used instead of totals	LSTM, BiLSTM and GRU against the MLP	Macro-F1
Rule features replaced by random features	Features named in the rule, top SHAP features and random features	Fall in the model's probability, share of changed predictions
Evidence removed from the feedback prompt	The five versions in Table 8	Share of messages with an incorrect claim
Check removed from the feedback	Grounded feedback with and without the claim check	Share of messages with an incorrect claim

The ablation is run for the MLP and the random forest, so that the same changes can be read for a neural model and for a machine-learning model.

4. Results

All results are for the 2014 test students (10,740), who were not used for training or tuning. Scores are reported as macro-F1 (the average of the F1 scores of the four classes) unless stated otherwise.

4.1. How well do the models predict? (RQ1)

All seven models predicted substantially better than guessing. Always guessing "Pass" gave a macro-F1 of 0.170, and the models gave 0.455 to 0.508 (Table

10). The random forest scored highest on macro-F1 (0.508), followed by XGBoost (0.502). Logistic regression and the MLP both scored 0.492. The three recurrent networks (LSTM, BiLSTM and GRU) scored lower (0.455 to 0.466), although these models use the weekly activity of each student. At the halfway point, the weekly pattern added nothing beyond the totals that the other models already use. Logistic regression had the highest balanced accuracy (0.547), with the MLP next (0.539).

Table 10 Prediction scores on the 2014 test set

Model	Macro-F1	Balanced accuracy	AUC
Always guess "Pass"	0.170	0.250	0.500
Random Forest	0.508	0.518	0.814
XGBoost	0.502	0.513	0.800
Logistic regression	0.492	0.547	0.816
MLP	0.492	0.539	0.812
GRU	0.466	0.499	0.790
LSTM	0.465	0.504	0.788
BiLSTM	0.455	0.499	0.786

The differences among the top four models are small, and this research does not claim that the MLP predicts better than the random forest overall. The two models make different kinds of correct predictions (Table 11). The MLP identified more of the smaller classes: Withdrawn students (35% against 15%) and Distinction students (72% against 58%). The random forest identified more of the

larger classes: Pass students (68% against 53%) and Fail students (67% against 56%). For an early-warning system, finding students who will withdraw is the harder and more useful task, and this is where the MLP is stronger. Table 11 shows Share of Each Outcome Identified by Random Forest and MLP (Recall)

Table 11 Share of Each Outcome Identified by Random Forest and MLP (Recall)

Outcome	Share of students	Random forest	MLP
Pass	52%	0.68	0.53
Fail	27%	0.67	0.56
Distinction	13%	0.58	0.72
Withdrawn	8%	0.15	0.35

The confusion matrix of the tuned random forest on the 2014 test set is given in Table 12. Each row shows how the students of one true outcome were

classified. Table 12 shows Confusion Matrix of the Tuned Random Forest On The 2014 Test Set

Table 12 Confusion Matrix of the Tuned Random Forest On The 2014 Test Set

True outcome Predicted	Distinction	Fail	Pass	Withdrawn	Total	Recall
Distinction	832	36	562	10	1,440	0.58
Fail	63	1,905	662	233	2,863	0.67
Pass	823	781	3,787	165	5,556	0.68
Withdrawn	27	476	249	129	881	0.15
Precision	0.48	0.60	0.72	0.24	10,740	-

The matrix shows where the errors of the random forest lie. Most errors are between neighbouring outcomes. Of the Distinction students, 39% (562 of 1,440) were predicted to pass, and 15% of the Pass students (823 of 5,556) were predicted to gain a Distinction. Fail and Pass are also confused in both directions (662 and 781 students). Withdrawn is the hardest outcome: only 129 of 881 Withdrawn

students (15%) were identified, and 54% (476) were predicted to fail. Withdrawn students resemble failing students in the activity and assessment records available at the halfway point, which explains why the recall for this smallest class is the lowest, and why only 24% of the students predicted to withdraw actually did so. Table 13 shows Confusion Matrix of Tuned MLP on 2014 Test Set

Table 13 Confusion Matrix of Tuned MLP on 2014 Test Set

True outcome Predicted	Distinction	Fail	Pass	Withdrawn	Total	Recall
Distinction	1,036	44	335	25	1,440	0.72
Fail	92	1,596	485	690	2,863	0.56
Pass	1,230	749	2,935	642	5,556	0.53
Withdrawn	49	376	148	308	881	0.35
Precision	0.43	0.58	0.75	0.19	10,740	-

Table 13 shows that the MLP is the more sensitive model for early warning. It identified 35% of the Withdrawn students against 15% for the random forest, and 79% of the students who failed or withdrew against 73%. Its balanced accuracy was also higher (0.539 against 0.518). The cost is more false alarms, with 21% of students who needed no support flagged against 14%, but a missed student is generally the costlier error. The MLP is therefore the more suitable model for identifying at-risk students, even though the random forest remains better overall by macro-F1.

4.2. Did tuning improve performance?

Tuning helped most for the random forest and the MLP (Table 13). The random forest rose from 0.424 to 0.508 and the MLP from 0.455 to 0.492. Logistic regression did not change, and XGBoost changed very little. Before tuning, the random forest identified only 2% of Withdrawn students. After tuning it identified 15%, and the MLP identified 35%. The cross-validation scores on the 2013 data (0.504 to 0.540) were slightly higher than the 2014 test scores for all four models. A small drop is likely when the test year is a new cohort. Table 14 shows Scores Before and After Tuning

Table 14 Scores Before and After Tuning

Model	Macro-F1 before tuning	Macro-F1 after tuning	Balanced accuracy before tuning	Balanced accuracy after tuning	Withdrawn found before tuning	Withdrawn found after tuning	Cross-validation macro-F1
Logistic Regression	0.492	0.492	0.547	0.547	0.38	0.38	0.504
Random Forest	0.424	0.508	0.431	0.518	0.02	0.15	0.540
XGBoost	0.499	0.502	0.503	0.513	0.16	0.17	0.533
MLP	0.455	0.492	0.498	0.539	0.30	0.35	0.508

4.3. Can simple rules approximate the model? (RQ2)

A small decision tree was trained to reproduce the predictions of each model. Each tree gives 15 or 16 if-then rules, for example: if the student has

submitted more than 83% of the assignments due, and the average score is above 86, the model predicts Distinction. The results below are for the random forest and the MLP. Table 15 shows Tests of The Rules

Table 15 Tests of The Rules

Test	Random Forest	MLP
Agreement with the model	82%	72%
Agreement from guessing the most common answer	49%	36%
Agreement of a tree trained on the true results	72%	55%
Overlap of chosen features after resampling (1 = identical)	0.91	0.85
Rules learned in 2013, tested in 2014	82%	72%
Rules learned in 2014, tested in 2014	83%	73%
Top five features shared with SHAP	5 of 5	3 of 5

The rule tree agreed with the random forest for 82% of students and with the MLP for 72% (Table 14). Guessing the most common prediction would agree for only 49% and 36%. A tree trained on the true

results, without the model, agreed less often (72% for the random forest and 55% for the MLP). The rules therefore describe the model, and not only the data. The MLP is harder to approximate than the

random forest, and 28% of its predictions are not captured by four levels of rules. When the training students were resampled 30 times, the trees chose almost the same features (overlap 0.91 for the random forest and 0.85 for the MLP). Rules learned from the 2013 students worked almost as well on the 2014 students as rules learned from 2014 students (82% against 83% for the random forest; 72% against 73% for the MLP). The rules shared all five top features with SHAP for the random forest and three of five for the MLP. A tree trained on the true results alone predicted much worse than the models (macro-F1 0.41, against 0.492 for the MLP and 0.508 for the random forest). An interpretable tree therefore cannot replace the model. It can only describe it. For 100 students per model, the features

named in the student's rule were replaced by the median of the training students (Table 15). The model's prediction fell by 0.195 for the MLP and 0.160 for the random forest. Replacing the same number of random behavioural features reduced it by only 0.058 and 0.044. The rules therefore name features that influence the model. The model's own top three SHAP features have somewhat greater influence (0.243 and 0.252), so the rules capture much of the model's evidence but not all of it. Table 16 shows Effect On the Model's Probability of Replacing Features with The Median Table 17 shows Share of Messages with an Incorrect Claim for Each Feedback Version (100 Messages Per Model; 95% Confidence Intervals in Brackets)

Table 16 Effect On the Model's Probability of Replacing Features with The Median

Features replaced	MLP	Random forest
Features named in the student's rule	0.195	0.160
Top three SHAP features	0.243	0.252
Random behavioural features (same number as the rule)	0.058	0.044

4.4. Is the feedback true to the evidence? (RQ3)

A language model wrote feedback for 100 students per model. A program then checked every claim against the student's real data. A message counted as

incorrect if at least one claim named a missing feature, gave the incorrect direction (above or below the class median), gave a value more than 5% off, or gave the incorrect risk level (Table 16).

Table 17 Share of Messages with an Incorrect Claim for Each Feedback Version (100 Messages Per Model; 95% Confidence Intervals in Brackets)

Information given	MLP	Random Forest
The student's values only	10% (5.5 to 17.4)	7% (3.4 to 13.7)
The student's values and the three most influential features (SHAP only)	0% (0.0 to 3.7)	0% (0.0 to 3.7)
The student's values and the rule (rule only)	28% (20.1 to 37.5)	25% (17.5 to 34.3)
Values, features and rules (grounded)	3% (1.0 to 8.5)	12% (7.0 to 19.8)
Values, features, rules and the check (grounded and verified)	0% (0.0 to 3.7)	0% (0.0 to 3.7)

When the language model saw only the student's values, 10% of messages (MLP) and 7% (random forest) contained an incorrect claim. When it was also given the students' three most influential

features, none did. Because the same students were used in both versions, the difference can be tested directly. All 10 % incorrect messages for the MLP and all 7% for the random forest became correct,

and none went the other way (exact McNemar test, $p = 0.002$ and $p = 0.016$). When the language model was given only the rule, the share rose to 28% and 25%. Every one of these errors was an incorrect “above” or “below” claim. The rule gives a cut-off, and the check compares claims with the class median, so a claim that follows the rule can still be marked incorrectly. With both the features and the rule, 3% and 12% of messages were incorrect, and a second attempt after the check removed all incorrect claims. The main gain therefore comes from giving the model’s most influential features, which is the SHAP evidence. The rule alone makes the feedback worse by this check, and adding the rule to the features did not reduce errors further. The check removes the remaining errors, but the messages that it rejected twice are replaced by a fixed template. This research does not claim that the rule layer reduces incorrect claims. LLM as a judge and the claim check agreed poorly (Table 17). For

contradicted claims, kappa was 0.155 for the random forest and -0.021 for the MLP. For unsupported claims, kappa was -0.052 and -0.063. The judge identified 2 of the 19 messages that the claim check marked incorrect for the random forest, and none of the 13 for the MLP. It also marked about 30% of the other messages as containing unsupported statements, most likely general advice that the evidence does not list. Its review also did not find fewer unsupported statements with the features and the rule (29% without, 34% with). The two checks do not measure the same thing: the claim check is exact and tests each statement against the stored values, while the judge applies a looser standard. The claim check is therefore the main measure, and the judge is a secondary check only. Table 18 shows Agreement Between the Claim Check and The Language-Model Judge

Table 18 Agreement Between the Claim Check and The Language-Model Judge

Model	Messages marked incorrect by the claim check	Judge marks contradicted	Kappa (contradicted)	Judge marks unsupported	Kappa (unsupported)
Random forest	19 of 300	2	0.155	3	-0.052
MLP	13 of 300	0	-0.021	1	-0.063

4.5. Ablation study

Each row of Table 19 changes one part of the system and shows what was added or removed (the design

is in Table 9). R Table 19 shows Results of the Ablation Study

Table 19 Results of the Ablation Study

Ablation Component	What was compared	Result
Learning model removed	A tree trained on the true results, against the models	Macro-F1 0.410, against 0.492 (MLP) and 0.508 (random forest)
Rules fitted to the true results, not to the model’s predictions	Agreement with the model	Random forest 72%, against 82% for rules fitted to the model; MLP 55%, against 72%
Rule layer removed	The model with SHAP only	Same prediction scores, but no interpretable rules and no measure of how closely the rules approximate the model

Tuning removed	Macro-F1 before and after	Random forest 0.424 to 0.508; MLP 0.455 to 0.492
Weekly sequence used instead of totals	LSTM, BiLSTM and GRU against the MLP	0.455 to 0.466, against 0.492
Rule features replaced by random features	Fall in the model's probability	MLP 0.058 for random features, against 0.195 for rule features; random forest 0.044, against 0.160
Key features removed from the feedback prompt	Messages with an incorrect claim	0% rises to 10% (MLP) and 7% (random forest)
Rule added to the key features	Messages with an incorrect claim	0% rises to 3% (MLP) and 12% (random forest)
Rule used without the key features	Messages with an incorrect claim	28% (MLP) and 25% (random forest)
Check removed from the feedback	Messages with an incorrect claim	0% rises to 3% (MLP) and 12% (random forest)

The ablation shows that the model cannot be dropped, that the rules explain it well only when fitted to its own predictions, and that the key features, not the rule, carry most of the feedback's benefit. Second, the rule tree must be fitted to the model's own predictions to explain the model well. Third, in the feedback, the key features carry most of the benefit. The rule adds an interpretable reason, but not accuracy, and used alone it makes the feedback worse by the claim check. Fig. 2 shows how the depth of the rule tree affects agreement with the model. Agreement of the surrogate climbs from 0.67 at depth 2 to 0.72 at depth 4 and reaches about

0.75 at depth 5, after which it levels off at 0.76 by depth 10. Limiting the tree to four levels therefore keeps the rules short for a loss of roughly four points of agreement. At every depth the surrogate matches the model more closely than a tree fitted to the true outcomes, which supports fitting the rules to the model's own predictions. Its accuracy against the true outcomes does not improve with depth (0.59 at depth 2 and 0.55 at depth 10), which is expected of a tree that follows the model, including its errors, rather than the data. Figure 2 shows Effect of The Depth of the Rule Tree for The MLP

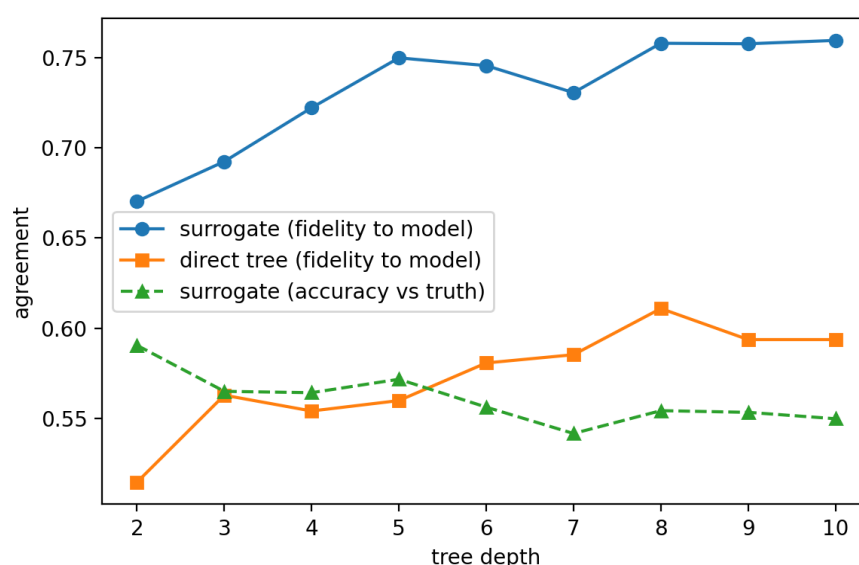


Figure 2 Effect of The Depth of the Rule Tree for The MLP

4.6. Comparison of the Random Forest and the MLP with Rules

The study compares a machine-learning model with its rules (random forest and symbolic layer) and a neural model with its rules (MLP and symbolic

layer). Table 19 puts all the measures side by side. Table 20 shows Random Forest with Rules vs MLP with Rules

Table 20 Random Forest with Rules vs MLP with Rules

Measure	Random forest and rules	MLP and rules	Stronger
Macro-F1	0.508	0.492	Random forest
Balanced accuracy	0.518	0.539	MLP
AUC	0.814	0.812	Similar
Pass found (recall)	0.68	0.53	Random forest
Fail found (recall)	0.67	0.56	Random forest
Distinction found (recall)	0.58	0.72	MLP
Withdrawn found (recall)	0.15	0.35	MLP
Rules agree with the model	82%	72%	Random forest
Agreement of a tree trained on the true results	72%	55%	Random forest
Overlap of chosen features after resampling	0.91	0.85	Random forest
Rules learned in 2013, tested in 2014	82%	72%	Random forest
Top five features shared with SHAP	5 of 5	3 of 5	Random forest
Fall in probability when rule features are replaced	0.160	0.195	MLP
Fall when the same number of random features are replaced	0.044	0.058	Similar
Incorrect claims, student's values only	7%	10%	Random forest
Incorrect claims, with SHAP features	0%	0%	Same
Incorrect claims, with the rule only	25%	28%	Similar
Incorrect claims, with features and rule	12%	3%	MLP
Incorrect claims, with features, rule and check	0%	0%	Same

Neither model with its rules is stronger on every measure. The random forest has the higher macro-F1, and its rules approximate it more closely, are more stable and agree more with SHAP. The MLP identifies more students in the two smallest classes, including Withdrawn students, has a slightly higher balanced accuracy, and had fewer incorrect claims when both the features and the rule were given (3% against 12%). The two models are equal on the feedback version with SHAP features and on the

verified version (0% incorrect claims). These differences come from one test year and 100 students per model for the feedback, and no significance test was run on these differences. This research therefore does not rank the two models. It shows that the same symbolic layer and the same feedback procedure work for both. The results show that each part does a different job. A model alone gives a prediction and no account of it. The rule trees supply that account: the trees agree with the

model for 82% of random forest and 72% of MLP predictions, against 49% and 36% for always predicting the most common class, remain stable under resampling and carry over from 2013 to 2014. The features named in the rules are ones the model relies on, since replacing these features lowers its confidence about three times as much as replacing random features. The feedback then depends on the model's evidence and the claim check, which bring the share of incorrect claims from 7% (random forest) and 10% (MLP) with the student's values alone to 0%. Taking a part away weakens the result. Without the model, a tree fitted to the true outcomes predicts worse (macro-F1 0.41), and the rule used without the model's evidence gives the most incorrect claims of any version (25% and 28%). The gain in reliability therefore comes mainly from the model's evidence and the check, while the symbolic layer provides a readable account of the model that can be tested.

5. Discussion

The three research questions received different answers. For RQ1, the neural models did not outperform the machine-learning models. The random forest scored highest (macro-F1 0.508) and the MLP matched logistic regression (0.492), but the two differ in the students identified: the MLP finds more Withdrawn students (35% against 15%), whereas the random forest finds more Pass and Fail students. The choice between the two depends on whether missing a student who withdraws costs more than a false alarm. For RQ2, small rule trees describe a trained model with useful accuracy. The rules agreed with the random forest for 82% of students and with the MLP for 72%, remained stable under resampling and carried over from 2013 to 2014. The features named in the rules influence the model, since replacing these features lowers its confidence about three times as much as replacing random features. The rules are not a full account of the model, however. The rules miss 28% of the MLP's predictions, and a tree fitted only to the true outcomes predicts much worse (macro-F1 0.41), so the rules describe the model and do not replace it. For RQ3, feedback was more reliable when the language model has given the model's most influential features. Incorrect claims fell from 10% (MLP) and 7% (random forest) to none. The rule on its own did not help. The gain in reliability therefore came from the model's evidence and the claim

check, while the symbolic layer supplied a compact account of the model that can be evaluated against data. Read together, the results favour the neuro-symbolic combination: the model predicts, the rules explain, and the evidence with the check keeps the feedback true to the data. The study rests on one dataset, one prediction time (50% of module length) and one language model, GPT-4.1, which wrote and judged the feedback without any other comparative study.

Conclusion and Future Work

A neuro-symbolic pipeline was tested for early-warning feedback on the Open University Learning Analytics Dataset. Seven models were trained on 2013 students and tested on 2014 students, with predictions made at the halfway point of each module, and all exceeded the majority-class baseline (macro-F1 0.455 to 0.508 against 0.170). Small rule trees fitted to the models agreed with the random forest for 82% of students and with the MLP for 72% and remained stable and carried over between years. Feedback grounded in the model's evidence and verified against the student's data contained no incorrect claims, against 10% (MLP) and 7% (random forest) for feedback written from the student's values alone. The random forest gave the best predictions (Macro-F1 0.508 against 0.492 for the MLP), but prediction accuracy is not where the neuro-symbolic combination earns its place. Its contribution begins after the prediction is made, and it is held for both models. The rules reproduced the random forest's output for 82% of students and the MLP's for 72%, well above the 49% and 36% obtained by always predicting the most common class. Replacing the features named in the rules lowered the models' confidence by 0.160 and 0.195, against 0.044 and 0.058 for random features, which shows that the rules point to features the models used one. When the language model was given the model's evidence and the claim check was applied, incorrect claims fell from 7% (random forest) and 10% (MLP) to none. The MLP also found more Withdrawn students than the random forest (35% against 15%), which counts for a good deal in early warning. In short, the model makes the prediction, the rules give a compact account of it that can be tested, and the evidence together with the claim check keeps the feedback in line with the student's data. The gain in reliability came from the evidence and the check, not from the rules alone. The study

does not show that neural models predict better than the random forest, nor that students and teachers find the feedback more useful. Future work will bring in teacher review of the rules and the feedback, use the rules to improve the neural model, repeat the study on other datasets, prediction times and judge models, and report the fairness audit across student groups.

References

- [1]. J. He, J. Bailey, B. I. P. Rubinstein, and R. Zhang, "Identifying at-risk students in massive open online courses," in Proc. AAAI Conf. Artif. Intell., 2015, doi: 10.1609/aaai.v29i1.9471.
- [2]. G. Akçapınar, A. Altun, and P. Aşkar, "Using learning analytics to develop early-warning system for at-risk students," Int. J. Educ. Technol. High. Educ., vol. 16, 2019, doi: 10.1186/s41239-019-0172-z.
- [3]. M. Adnan et al., "Predicting at-risk students at different percentages of course length for early intervention using machine learning models," IEEE Access, 2021, doi: 10.1109/ACCESS.2021.3049446.
- [4]. J. Kuzilek, M. Hlosta, and Z. Zdrahal, "Open University learning analytics dataset," Sci. Data, vol. 4, 2017, Art. no. 170171, doi: 10.1038/sdata.2017.171.
- [5]. N. R. Aljohani, A. Fayoumi, and S.-U. Hassan, "Predicting at-risk students using clickstream data in the virtual learning environment," Sustainability, vol. 11, no. 24, p. 7238, 2019, doi: 10.3390/su11247238.
- [6]. N. Jha, I. Ghergulescu, and A.-N. Moldovan, "OULAD MOOC dropout and result prediction using ensemble, deep learning and regression techniques," in Proc. Int. Conf. Comput. Supported Educ. (CSEDU), 2019, doi: 10.5220/0007767901540164.
- [7]. A. Gupta, D. Garg, and P. Kumar, "Mining sequential learning trajectories with hidden Markov models for early prediction of at-risk students in e-learning environments," IEEE Trans. Learn. Technol., 2022, doi: 10.1109/TLT.2022.3197486.
- [8]. Z. Shou, M. Xie, J.-J. Mo, and H. Zhang, "Predicting student performance in online learning: A multidimensional time-series data analysis approach," Appl. Sci., vol. 14,
- [9]. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems, vol. 30, 2017.
- [10]. M. T. Ribeiro, S. Singh, and C. Guestrin, "“Why should I trust you?” Explaining the predictions of any classifier," in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, doi: 10.1145/2939672.2939778.
- [11]. V. Swamy, B. Radmehr, N. Krco, M. Marras, and T. Käser, "Evaluating the explainers: Black-box explainable machine learning for student success prediction in MOOCs," in Proc. Int. Conf. Educ. Data Mining, 2022, doi: 10.48550/arXiv.2207.00551.
- [12]. M. W. Craven and J. W. Shavlik, "Extracting tree-structured representations of trained networks," in Advances in Neural Information Processing Systems, vol. 8, 1996.
- [13]. D. Hooshyar, R. Azevedo, and Y. Yang, "Augmenting deep neural networks with symbolic educational knowledge: Towards trustworthy and interpretable AI for education," Mach. Learn. Knowl. Extr., vol. 6, no. 1, 2024, doi: 10.3390/make6010028.
- [14]. F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," arXiv:1702.08608, 2017.
- [15]. [15] S. Steinert, K. E. Avila, S. Ruzika, J. Kühn, and S. Küchemann, "Harnessing large language models to develop research-based learning assistants for formative feedback," Smart Learn. Environ., 2024, doi: 10.1186/s40561-024-00354-1.
- [16]. J. Stamper, R. Xiao, and X. Hou, "Enhancing LLM-based feedback: Insights from intelligent tutoring systems and the learning sciences," in AIED Companion, 2024, doi: 10.48550/arXiv.2405.04645.
- [17]. X. Meng and Y. Yang, "Enhancing automated feedback in intelligent tutoring systems via large language models and data distillation," in Proc. Int. Conf. Innovation Eng. Technol., 2025, doi: 10.1109/ICIET66371.2025.11046219.
- [18]. M. Steinbach, S. Bhandari, J. Meyer, and Z. Pardos, "When LLMs hallucinate:

- Examining the effects of erroneous feedback in math tutoring systems,” Zenodo, 2025, doi: 10.5281/zenodo.15870126.
- [19]. U. Maier, M. Seibold, and C. Klotz, “LLM-generated formative feedback in education: A qualitative systematic literature review,” *Comput. Educ. Open*, 2026, doi: 10.1016/j.caeo.2026.100374.
- [20]. B. Davazdahemami, H. M. Zolbanin, and D. Delen, “Explain, embed, retrieve, and reason (E2R2): A SHAP-informed LLM framework for decision support,” in *Proc. Hawaii Int. Conf. Syst. Sci. (HICSS)*, 2026, doi: 10.24251/hicss.2026.198.
- [21]. L. Zheng et al., “Judging LLM-as-a-judge with MT-Bench and Chatbot Arena,” in *Advances in Neural Information Processing Systems*, vol. 36 (Datasets and Benchmarks), 2023.
- [22]. N. S. Raj, S. Prasad, P. Harish, M. Boban, and N. Cheriyaedath, “Early prediction of at-risk students in a virtual learning environment using deep learning techniques,” in *Lecture Notes in Computer Science*, 2021, doi: 10.1007/978-3-030-77873-6_8.
- [23]. M. Jawthari and V. Stoffa, “Predicting at-risk students using weekly activities and assessments,” *Int. J. Emerg. Technol. Learn.*, vol. 17, no. 19, 2022, doi: 10.3991/ijet.v17i19.31349.
- [24]. P. M. Moreno-Marcos et al., “Temporal analysis for dropout prediction using self-regulated learning strategies in self-paced MOOCs,” *Comput. Educ.*, 2019, doi: 10.1016/j.compedu.2019.103728.
- [25]. K. Liu, S. Tatinati, and A. W. H. Khong, “A weighted feature extraction technique based on temporal accumulation of learner behavior features for early prediction of dropouts,” in *Proc. IEEE Int. Conf. Teaching, Assessment, and Learning for Engineering (TALE)*, 2020, doi: 10.1109/TALE48869.2020.9368317.
- [26]. D. Bañeres, M. E. Rodríguez, and A.-E. Guerrero-Roldán, “A real-time predictive model for identifying course dropout in online higher education,” *IEEE Trans. Learn. Technol.*, 2023, doi: 10.1109/TLT.2023.3267275.
- [27]. D. M. Olivé, D. Q. Huynh, M. Reynolds, M. Dougiamas, and D. Wiese, “A quest for a one-size-fits-all neural network: Early prediction of students at risk in online courses,” *IEEE Trans. Learn. Technol.*, vol. 12, no. 2, pp. 171–183, 2019, doi: 10.1109/TLT.2019.2911068.
- [28]. 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, doi: 10.1007/BF01099821.
- [29]. C. Piech et al., “Deep knowledge tracing,” in *Advances in Neural Information Processing Systems*, vol. 28, 2015.
- [30]. S. Pandey and G. Karypis, “A self attentive model for knowledge tracing,” in *Proc. Int. Conf. Educ. Data Mining*, 2019, arXiv:1907.06837.
- [31]. A. Ghosh, N. Heffernan, and A. S. Lan, “Context-aware attentive knowledge tracing,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, doi: 10.1145/3394486.3403282.
- [32]. B. Xu et al., “Learning behavior-oriented knowledge tracing,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2023, doi: 10.1145/3580305.3599407.
- [33]. S. Minn, J.-J. Vie, K. Takeuchi, H. Kashima, and F. Zhu, “Interpretable knowledge tracing: Simple and efficient student modeling with causal relations,” in *Proc. AAAI Conf. Artif. Intell.*, 2022, doi: 10.1609/aaai.v36i11.21560.
- [34]. Y. Liang, T. Peng, Y. Pu, and W. Wu, “HELP-DKT: An interpretable cognitive model of how students learn programming based on deep knowledge tracing,” *Sci. Rep.*, vol. 12, 2022, doi: 10.1038/s41598-022-07956-0.
- [35]. F. Liu, C. Bu, H. Zhang, L. Wu, K. Yu, and X. Hu, “FDKT: Towards an interpretable deep knowledge tracing via fuzzy reasoning,” *ACM Trans. Inf. Syst.*, 2024, doi: 10.1145/3656167.
- [36]. Y. Wang, Y. Huo, C. Yang, X. Huang, D. Xia, and F. Feng, “Knowledge ontology enhanced model for explainable knowledge tracing,” *J. King Saud Univ. Comput. Inf.*

- Sci., 2024, doi: 10.1016 /j.jksuci. 2024. 102065.
- [37]. M. K. J. Carlon and J. S. Cross, “Knowledge tracing for adaptive learning in a metacognitive tutor,” *Open Educ. Stud.*, 2022, doi: 10.1515/edu-2022-0013.
- [38]. J. M. Alonso and G. Casalino, “Explainable artificial intelligence for human-centric data analysis in virtual learning environments,” in *Higher Education Learning Methodologies and Technologies Online*, 2019, doi: 10.1007/978-3-030-31284-8_10.
- [39]. B. I. Smith, C. Chimedza, and J. H. Bührmann, “Individualized help for at-risk students using model-agnostic and counterfactual explanations,” *Educ. Inf. Technol.*, 2021, doi: 10.1007/s10639-021-10661-6.
- [40]. M. Nagy and R. Molontay, “Interpretable dropout prediction: Towards XAI-based personalized intervention,” *Int. J. Artif. Intell. Educ.*, 2023, doi: 10.1007/s40593-023-00331-8.
- [41]. B. Pei and W. Xing, “An interpretable pipeline for identifying at-risk students,” *J. Educ. Comput. Res.*, 2021, doi: 10.1177/073563312111038168.
- [42]. A. Bettahi, F.-Z. Belouadha, and H. Harroud, “A modular and explainable machine learning pipeline for student dropout prediction in higher education,” *Algorithms*, vol. 18, no. 10, 2025, doi: 10.3390/a18100662.
- [43]. M. Afzaal et al., “Explainable AI for data-driven feedback and intelligent action recommendations to support students self-regulation,” *Front. Artif. Intell.*, 2021, doi: 10.3389/frai.2021.723447.
- [44]. G. Casalino, G. Castellano, and G. Zaza, “Explainable fuzzy models for learning analytics,” in *Lecture Notes in Networks and Systems*, 2023, doi: 10.1007/978-3-031-35501-1_40.
- [45]. V. Swamy, S. Du, M. Marras, and T. Käser, “Trusting the explainers: Teacher validation of explainable artificial intelligence for course design,” in *Proc. Int. Conf. Learn. Analytics Knowl. (LAK)*, 2023, doi: 10.1145/3576050.3576147.
- [46]. A. d’Avila Garcez and L. C. Lamb, “Neurosymbolic AI: The 3rd wave,” *Artif. Intell. Rev.*, vol. 56, 2023, arXiv:2012.05876.
- [47]. K. Ruzimboev, I. D. Avezmatov, and A. Xo’jamqulov, “A neuro-symbolic, multi-modal architecture for advancing adaptive intelligent tutoring systems,” in *Proc. IEEE Int. Conf. Actual Problems of Electronics Instrument Engineering (APEIE)*, 2025, doi: 10.1109/APEIE66761.2025.11289439.
- [48]. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Belmont, CA, USA: Wadsworth, 1984.
- [49]. S. Wang, X. Luo, and K. Hew, “Enhancing self-regulated learning using generative AI: Development and evaluation of SRLMentor,” *IEEE Trans. Learn. Technol.*, 2025, doi: 10.1109/TLT.2025.3634216.
- [50]. J. Gardner, C. Brooks, and R. Baker, “Evaluating the fairness of predictive student models through slicing analysis,” in *Proc. Int. Conf. Learn. Analytics Knowl. (LAK)*, 2019, doi: 10.1145/3303772.3303791.
- [51]. A. Al-Zawqari and G. Vandersteen, “Fairness in predictive learning analytics: A case study in online STEM education,” in *Proc. IEEE Frontiers in Education Conf. (FIE)*, 2023, doi: 10.1109/FIE58773.2023.10343059.
- [52]. K. McConvey and S. Guha, “Designing for fairness in higher education early warning systems,” in *Proc. Canadian Conf. Artif. Intell.*, 2024, doi: 10.21428/594757db.c7aec3c1.
- [53]. T. Blazkova, R. F. Kizilcec, M. L. Nielsen, D. D. Lassen, and A. Bjerre-Nielsen, “Fairness over time: A nationwide study of evolving bias in dropout prediction,” in *Proc. ACM Conf. Learning @ Scale*, 2025, doi: 10.1145/3698205.3733933.
- [54]. L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [55]. T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.
- [56]. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [57]. M. Schuster and K. K. Paliwal,

- “Bidirectional recurrent neural networks,” IEEE Trans. Signal Process., vol. 45, no. 11, pp. 2673–2681, 1997.
- [58]. K. Cho et al., “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 1724–1734.
- [59]. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2015.
- [60]. F. Wilcoxon, “Individual comparisons by ranking methods,” Biometrics Bull., vol. 1, no. 6, pp. 80–83, 1945.
- [61]. J. Cohen, “A coefficient of agreement for nominal scales,” Educ. Psychol. Meas., vol. 20, no. 1, pp. 37–46, 1960.