AI-as-a-service Score Stacking for Low-infrastructure Analytics: Continuous LLM Scoring of Vietnamese Student Feedback

Huy-Minh Dau [1,2,*] and Thi-Phan-Thu Pham [1,2]

[1] University of Social Sciences and Humanities, Ho Chi Minh, Viet Nam
[2] Viet Nam National University, Ho Chi Minh, Viet Nam
Email: dauhuyminh@hcmussh.edu.vn (H.-M.D.); 241814011408@hcmussh.edu.vn (T.-P.-T.P.)
*Corresponding author

Manuscript received June 9, 2026; revised June 22, 2026; accepted July 28, 2026; published November 24, 2026

Abstract*—Many higher education institutions seek to use artificial intelligence for educational analytics but lack the infrastructure, expertise, or resources required to train, fine-tune, or host large language models locally. This study proposes an Artificial Intelligence (AI)-as-a-Service (AIaaS) score-stacking framework for Vietnamese student feedback sentiment analysis. Instead of treating Large Language Models (LLMs) as direct label predictors, the framework uses externally served LLMs as continuous sentiment scorers. Gemma-4-31B-IT, Llama-3.1-8B-Instant, and Qwen3-32B were accessed through AIaaS inference platforms to assign sentiment scores in the interval [−1, 1] to 16,169 sentences from the University of Information Technology-Vietnamese Students' Feedback Corpus (UIT-VSFC) corpus. These scores were transformed into raw-score, aggregate, disagreement, and interaction features, then used to train lightweight local decision models. The novelty of this approach lies in using externally served LLM outputs as continuous meta-learning signals rather than as final sentiment labels. The evaluation compared fixed-threshold baselines, tuned-threshold baselines, equal and weighted score aggregation, and probability-stacking classifiers. In this evaluation, probability-stacking models achieved higher macro-F1 than direct thresholding and simple score aggregation. The final model, ExtraTrees stacking with probability-bias tuning selected by development-set macro-F1, achieved a test macro-F1 of 0.790. The findings suggest that, in this setting, the value of multi-model AIaaS lies not in naive averaging but in locally calibrated decision learning over continuous LLM score signals. This study contributes a technically feasible and auditable architecture for low-infrastructure higher education analytics, where institutions can exploit external LLM capabilities while retaining local control over calibration, evaluation, and decision-making.

***Keywords*—AI-as-a-service, student feedback analytics, sentiment analysis, score stacking, higher education**

I. Introduction

Student feedback is one of the most important textual data sources for understanding teaching quality, learner experience, course design, and institutional service improvement in higher education. Unlike structured rating scales, open-ended feedback allows students to express concrete concerns, contextual explanations, suggestions, and affective evaluations in their own words. However, the same openness that makes student feedback valuable also makes it difficult to analyze at scale. As digital learning environments and institutional quality assurance systems generate increasing volumes of learner comments, higher education institutions require analytic methods that can transform textual feedback into usable evidence for teaching improvement and educational management. Educational data mining and learning analytics have therefore become important approaches for supporting monitoring, intervention, quality enhancement, and institutional decision-making [1].

Within this broader analytics agenda, sentiment analysis has been widely used to extract affective and evaluative signals from student comments. Prior studies and reviews have examined sentiment analysis in higher education, MOOCs, online learning environments, course evaluations, and formative assessment, showing that sentiment classification can help institutions summarize large-scale feedback and identify patterns of satisfaction or dissatisfaction [2–4]. In the Vietnamese context, the University of Information Technology-Vietnamese Students' Feedback Corpus (UIT-VSFC) corpus has provided an important benchmark for student feedback sentiment analysis, enabling research on traditional machine learning, recurrent neural networks, deep learning models, lexicon-enhanced approaches, and transformer-based classifiers [5, 6]. These studies have established the feasibility of supervised sentiment classification for Vietnamese student feedback.

Nevertheless, existing approaches leave open an important implementation problem. Much prior work asks which text classification model can perform well on student feedback. This is a valid modelling question, but it does not fully address the institutional question faced by many higher education institutions: how can a university operationalize Artificial Intelligence (AI)-based feedback analytics when it lacks the infrastructure, expertise, or resources required to train, fine-tune, or host large language models locally? In this study, low-infrastructure refers primarily to limited local computational capacity, limited AI engineering expertise, and limited institutional capacity to maintain large-model training or hosting pipelines, rather than to the absence of educational data. This constraint is consistent with recent institutional Large Language Models (LLM) platform studies, which show that university deployment requires secure inference, access control, cost management, Application Programming Interface (API) integration, data protection, and long-term maintenance mechanisms [7, 8]. This question is increasingly important because the adoption of generative AI and large language models in higher education is not only a matter of technical performance. It is also shaped by technological readiness, privacy, transparency, equity, stakeholder training, governance, and institutional accountability [9–11].

Large language models create a new opportunity to address this problem because they can be accessed through AI-as-a-Service (AIaaS) platforms. Instead of building the entire AI stack locally, institutions can access externally served models through cloud-based interfaces or APIs. Recent institutional LLM platform initiatives have explored service-oriented architectures that combine commercial APIs, cloud-hosted models, and locally managed open models while providing access control, privacy mechanisms, budget management, retrieval-augmented generation, and API access for university settings [7, 8]. These developments suggest that higher education institutions may increasingly consume AI capability as a service rather than develop every component internally.

However, simply calling an external LLM and accepting its predicted label is not sufficient for responsible educational analytics. External models may differ in calibration, score distribution, output stability, sensitivity to language, and behavior across ambiguous cases. Their outputs may also be misaligned with local datasets, class distributions, and institutional analytic goals. This is especially problematic in student feedback sentiment analysis, where comments may be short, mixed, descriptive, weakly affective, or context-dependent. The neutral class is particularly challenging because it often contains factual statements, mild suggestions, or ambiguous evaluations, and it is usually underrepresented relative to positive and negative comments.

This study proposes a different use of LLMs in educational sentiment analysis. In this paper, score stacking refers to the use of continuous LLM sentiment scores as intermediate inputs for locally trained decision models, whereas probability stacking refers specifically to the variant in which a local classifier produces class probabilities that are subsequently adjusted through development-set probability-bias tuning. Instead of treating LLMs as direct label predictors, the proposed framework treats them as continuous sentiment scorers. Each model assigns a sentiment score in the interval $[-1, 1]$, where negative values indicate negative sentiment, values near zero indicate neutral or weakly affective content, and positive values indicate positive sentiment. This design is grounded in the broader sentiment analysis literature, which recognizes that sentiment can be represented not only as discrete polarity classes but also as polarity intensity or continuous affective orientation [12–14]. The final sentiment label is then predicted by a lightweight local decision layer trained on score-derived features.

The methodological foundation of this design is stacked generalization. Ensemble learning research shows that multiple models can be combined in different ways, from hard voting and soft voting to weighted aggregation and stacking. Stacking is particularly relevant because it trains a higher-level model to combine the outputs of lower-level models rather than simply averaging or voting among them [15]. For classification tasks, prior work has emphasized that confidence or probability outputs are more informative inputs to the meta-learner than hard predicted labels [16]. In the present study, continuous LLM sentiment scores play a similar role: they are not final decisions, but intermediate signals that can be calibrated and combined by a local meta-classifier. Unlike conventional ensemble learning, which usually combines locally trained classifiers, the proposed approach combines continuous score signals produced by externally served LLMs and restricts local learning to a lightweight decision layer.

Building on these ideas, this study introduces and evaluates an AIaaS score-stacking framework for Vietnamese student feedback sentiment analysis. Three LLMs—Gemma-4-31B-IT, Llama-3.1-8B-Instant, and Qwen3-32B—were accessed through AIaaS inference platforms and used to generate continuous sentiment scores for the UIT-VSFC dataset. These scores were transformed into raw-score, aggregate, disagreement, and interaction features. A set of lightweight local decision models was then evaluated, including fixed-threshold baselines, tuned-threshold baselines, equal and weighted score aggregation, and probability-stacking classifiers.

The study addresses the following research question: Can multiple externally served LLMs be used as continuous sentiment scorers and combined through a lightweight local decision layer to provide a feasible and empirically evaluable approach to student feedback sentiment classification under low-infrastructure conditions?

The study makes four contributions. First, it reframes LLM use in student feedback analytics from direct label prediction to continuous semantic scoring for local meta-learning. Second, it proposes an AIaaS score-stacking architecture that separates external semantic interpretation from local decision calibration, evaluation, and model selection. Third, it provides an empirical evaluation on UIT-VSFC, comparing single-model scoring, score aggregation, and probability-stacking methods under a development-based model selection protocol. Fourth, it identifies implementation considerations associated with this architecture for higher education institutions that need AI-supported analytics but may not have the capacity to train, fine-tune, or host large models locally.

II. Related Work

A. Student Feedback Sentiment Analysis and Continuous Sentiment Representation

Student feedback constitutes a valuable source of evidence for understanding teaching quality, learning experience, course design, and institutional service improvement. In higher education, textual feedback is particularly important because it often captures learners' perceptions in a more open-ended and context-sensitive manner than structured rating scales. However, the increasing volume of digital feedback makes manual analysis difficult to scale, especially when institutions need timely evidence for quality assurance, curriculum improvement, and educational management. This has motivated the use of educational data mining and learning analytics techniques to transform learner-generated data into actionable information for instructors and institutional managers [1].

Within this broader analytics agenda, sentiment analysis has become a common approach for extracting affective and evaluative signals from student comments. Prior reviews show that sentiment analysis has been widely applied to educational contexts, including online learning environments, MOOCs, course evaluation, and formative

assessment [2–4]. These studies generally treat student feedback as a textual data source from which positive, negative, and neutral orientations can be inferred. A related fine-grained line of work is aspect-based sentiment analysis, which seeks to identify the educational aspects being evaluated and the sentiment associated with those aspects; recent studies on teacher-performance evaluation show that Aspect-Based Sentiment Analysis (ABSA) can convert open-ended student comments into more targeted evidence about instructional strengths and areas for improvement [17, 18]. Such classification is useful because it enables institutions to summarize large volumes of learner comments, identify problematic learning experiences, and monitor patterns of satisfaction or dissatisfaction across courses and cohorts.

Vietnamese student feedback sentiment analysis has also emerged as a specific research context. The UIT-VSFC corpus provides a benchmark dataset of Vietnamese student feedback annotated for sentiment and topic classification, thereby enabling systematic experimentation on educational sentiment analysis in Vietnamese [5]. Subsequent studies have applied traditional machine learning, deep learning, recurrent neural networks, transformer-based models, and lexicon-enhanced methods to this corpus, indicating that UIT-VSFC has become an important reference dataset for Vietnamese educational text analytics [19, 20]. Nevertheless, the dataset also reflects a common methodological challenge in educational sentiment analysis: the neutral class is much smaller than the positive and negative classes, making class-balanced evaluation and decision calibration especially important.

Most sentiment analysis studies operationalize sentiment as a discrete classification problem. In the simplest case, a text is assigned to positive or negative polarity; in more educationally relevant settings, a neutral class is added to distinguish evaluative comments from descriptive or ambiguous statements. However, sentiment is not only categorical. The broader sentiment analysis literature recognizes that polarity can also be represented by intensity, where a text may express different degrees of positivity or negativity rather than a fixed class label [13, 14]. In this view, polarity may be modeled on an ordinal or continuous scale, with negative values indicating negative sentiment, positive values indicating positive sentiment, and values close to zero representing neutrality or weak affective orientation. This perspective is particularly relevant to student feedback, where short, mixed, or weakly affective comments may benefit from an intermediate continuous score before final class calibration.

Recent work in affective computing and sentiment analysis supports this broader view of sentiment representation. Emotion and sentiment intensity prediction studies have shown that affective meaning can be modeled not only through coarse-grained categories but also through fine-grained continuous or ordinal dimensions such as intensity, valence, and arousal [12]. Such approaches are particularly relevant when the objective is not merely to classify text, but to preserve more nuanced affective signals for downstream decision-making. For student feedback analytics, this suggests that large language models need not be used only as direct label predictors. They can instead serve as semantic scorers that map each feedback sentence onto a continuous sentiment scale.

B. LLMs and AI-as-a-Service in Higher Education Analytics

The emergence of large language models has expanded the range of educational tasks that can be supported by artificial intelligence. Because many educational processes depend on textual data, LLMs are particularly relevant to higher education settings where institutions routinely generate course evaluations, student feedback, advising records, learning materials, administrative inquiries, and policy documents. Recent reviews suggest that LLM-based educational technologies have been explored for a wide range of tasks, including content generation, feedback provision, grading, teaching support, recommendation, profiling, labelling, and detection [11, 21]. More broadly, AI-supported educational technologies also extend beyond text-only systems, including immersive and adaptive learning environments that integrate artificial intelligence with virtual reality to support learning processes [22]. More directly related to educational feedback analytics, Parker *et al.* [23] showed that GPT-based LLM workflows can support analysis of education feedback surveys through tasks such as classification, extraction, thematic analysis, and sentiment analysis, illustrating the potential of LLMs for deriving insight from open-ended course-evaluation comments. Recent Vietnamese sentiment studies provide a more direct point of comparison. Thin *et al.* [24] evaluated zero-shot and few-shot prompting across GPT-family models, Llama-3, and SeaLLM on six Vietnamese sentiment datasets, including UIT-VSFC. Their results showed that few-shot examples could improve some model-dataset combinations, but the effects were not uniform, and fine-tuned models such as viT5, XLM-RoBERTa, and PhoBERT remained stronger on UIT-VSFC. More recently, Phuc *et al.* [25] compared zero-shot, few-shot, chain-of-thought, and knowledge-aided prompting across Qwen, Gemma, Llama, and GPT models on the UIT-VSFC test set, reporting substantial variation in macro-F1, latency, and cost across models and prompt configurations. Together, these studies indicate that direct categorical prompting is an important baseline, but its effectiveness remains sensitive to model choice, prompt design, and evaluation conditions. These developments indicate that LLMs are not merely conversational tools, but increasingly function as general-purpose language infrastructures for educational analytics and institutional services.

However, the adoption of LLMs in higher education should not be understood solely as a matter of model capability. It is also an institutional implementation problem. Prior work has highlighted practical and ethical challenges associated with LLM-based educational innovations, including technological readiness, replicability, transparency, privacy, equality, and beneficence [11]. At the policy level, universities have also had to develop guidelines for generative AI adoption, addressing issues such as academic integrity, teaching and learning practices, equity, data privacy, stakeholder training, and institutional responsibility [10]. These concerns suggest that the question is not simply whether LLMs can perform a task, but how higher education institutions can deploy, govern, and adapt

them in sustainable and accountable ways.

This implementation issue is especially important for institutions with limited in-house AI infrastructure. Training, fine-tuning, and hosting large language models locally may require substantial computational resources, specialized engineering expertise, data governance mechanisms, and long-term maintenance capacity. Even when open-source LLMs are available, local deployment involves non-trivial decisions about model selection, Graphics Processing Unit (GPU) resources, inference optimization, security, monitoring, and integration with existing information systems. Institutional LLM platform studies have therefore emphasized the importance of secure inference, access control, role-based authentication, API compatibility, data isolation, and cost-aware resource management [7]. These requirements are often beyond the immediate capacity of many higher education units, especially those outside computer science or engineering domains. In this context, AI-as-a-Service offers a service-oriented alternative in which computationally intensive semantic interpretation can be delegated to external inference platforms, while the institution retains control over task design, evaluation, and local calibration.

Recent institutional AI platform initiatives point in this direction. For example, university-oriented LLM platforms have explored hybrid infrastructures that combine commercial APIs, cloud-hosted models, and locally managed open models, while providing user interfaces, access control, budget management, retrieval-augmented generation, and API endpoints for institutional use [8]. Similarly, secure self-service LLM exploration platforms have been designed to make customized LLM services accessible to institutional users while addressing data protection, model isolation, encryption, and integration concerns [7]. These studies demonstrate that LLM adoption in higher education is increasingly moving from isolated experimentation toward platform-based and service-oriented architectures.

Nevertheless, AI-as-a-Service still requires institutional adaptation, because external model outputs may be biased, unstable, or insufficiently aligned with local analytic goals. From a calibration perspective, recent work on LLM confidence estimation emphasizes that LLM outputs should not be treated as inherently reliable confidence signals; instead, confidence-like outputs require task-specific evaluation, calibration, and uncertainty-aware interpretation before being used in downstream decision pipelines [26]. For educational feedback analytics, this supports treating LLMs as semantic scoring services rather than final decision-makers.

This perspective reframes AI-as-a-Service from a passive consumption model into an orchestration model. The institution does not merely consume an AI output; it designs a pipeline in which multiple externally served LLMs produce intermediate semantic signals, and a local decision layer learns how to combine those signals for a specific educational analytics task. Such a design is aligned with the broader management logic of learning analytics and educational data mining, where the objective is not only to produce predictions, but to transform educational data into evidence that can support monitoring, intervention, quality improvement, and institutional decision-making [1].

C. Ensemble Learning, Soft Voting, and Stacked Generalization

Ensemble learning is motivated by the idea that different models may capture different aspects of the data and make different errors. In classification tasks, common ensemble strategies include hard voting, soft voting, weighted aggregation, bagging, boosting, and stacking. Voting and score aggregation are simple to implement, but they may discard uncertainty, ignore class-specific reliability [27], or compress informative disagreement into a single combined score. This limitation is especially relevant in multi-LLM settings, where models may differ in calibration, confidence, and sensitivity to ambiguous language.

Stacked generalization provides a more flexible alternative because it trains a higher-level model to interpret the outputs of lower-level models rather than selecting a single model or applying a fixed voting rule. Wolpert [15] introduced stacking as a method for reducing generalization error by learning from the behavior of multiple generalizers on held-out data.

A key issue in stacking is what information should be passed to the meta-learner. Ting and Witten [16] showed that stacking for classification benefits from using confidence or class probability outputs rather than only predicted labels. This is directly relevant to the present study because continuous sentiment scores preserve information about intensity, uncertainty, and proximity to decision boundaries.

In a multi-LLM setting, this distinction becomes especially important. LLMs may differ in training data, model scale, alignment, and inference behavior; therefore, their outputs can be treated either as final labels to be combined or as intermediate semantic signals for a downstream decision model. The latter interpretation is more consistent with stacking. For student feedback sentiment analysis, treating LLM scores as intermediate signals is useful because comments may be short, elliptical, mixed, or institution-specific. Model disagreement can therefore be informative, and derived features such as score mean, minimum, maximum, range, and absolute disagreement may help the local decision layer identify ambiguous or boundary cases.

D. Evaluation, Calibration, and Class Imbalance

Evaluation metric choice is especially consequential in class-imbalanced sentiment analysis. A model may achieve high accuracy by performing well on majority classes while remaining weak on minority classes. In student feedback sentiment analysis, this is problematic because neutral comments, although less frequent, may contain descriptive, mixed, suggestive, or ambiguous information that remains relevant for educational decision-making.

The distinction between accuracy, weighted metrics, and macro-averaged metrics therefore requires explicit justification. Opitz [28] argues that classification evaluation is often insufficiently transparent, especially when macro-F1 is selected through vague references to class imbalance. In educational sentiment analysis, macro-F1 is relevant because it gives equal weight to each class and therefore makes minority-class weakness more visible than accuracy or weighted-F1.

Class imbalance also affects confidence calibration, that is,

the correspondence between model confidence and actual correctness. In imbalanced datasets, posterior probabilities and confidence estimates may become less reliable for minority classes. Recent work shows that class imbalance can distort posterior probability estimation and make calibration more difficult, especially for uncertain minority classes [29]. Raw model scores should therefore not be assumed to map directly onto final classes without task-specific calibration.

Decision calibration can be implemented through threshold tuning for continuous scores or probability adjustment for stacking models. Fixed thresholds are simple and transparent, but they assume that the score scale is already aligned with the target label distribution, which may be unrealistic for externally served LLMs. Development-set tuning can adapt thresholds or probability-bias terms to the empirical distribution and the selected evaluation objective, provided that the held-out test set is used only for final reporting.

III. Methodology

A. Ethical Approval and Informed Consent

This study used a secondary anonymized textual dataset previously released for research purposes. No human participants were recruited, contacted, interviewed, surveyed, observed, or experimentally involved by the authors, and no new personal or identifiable data were collected. According to the institutional regulations of the University of Social Sciences and Humanities, Viet Nam National University Ho Chi Minh City, ethics review applies to research involving human participants, defined as individuals who directly or indirectly participate in the research by providing information, being surveyed, observed, or experimentally involved. Since the present study did not involve such activities, it did not fall within the scope of human-subject research requiring ethics committee approval. Therefore, ethical approval and informed consent were not required.

All data were analyzed and reported in aggregated form, and no identifiable information about any individual participant is presented in this manuscript.

A separate ethical consideration concerns the use of external AIaaS services for textual scoring. In the present experiment, the inputs sent to the external scoring services were limited to the publicly released UIT-VSFC benchmark sentences, dataset record identifiers, and calibration anchors. The original UIT-VSFC construction procedure included a de-identification step in which person names were anonymized by replacement with symbolic placeholders [5]. Thus, the present study did not add student names, student numbers, contact information, course-enrolment records, or institutional administrative identifiers to the texts submitted for scoring. However, benchmark-level anonymization should not be treated as sufficient for real institutional deployment. The reported analyses did not depend on provider-side storage or model-improvement processes; all numerical results were reproduced from locally archived scorer-level outputs, whereas provider-side retention, logging, and model-improvement terms should be reviewed as deployment-level governance requirements before any non-public institutional feedback is submitted to external services. A production workflow should apply an explicit pre-submission de-identification pipeline, including named-entity detection for personal names, lecturer names, class or course identifiers, and organization names; regular-expression filtering for student IDs, email addresses, phone numbers, URLs, and dates; dictionary-based replacement of locally known teacher, course, and unit names; and manual review of rare-event or complaint-specific comments that may contain indirect identifiers. In addition, corpus licensing conditions, local access-control logs, and institutional approval rules would also need to be reviewed before any non-public student feedback is processed through external AIaaS services.

B. Research Design

This study employed a computational experimental design to evaluate an AIaaS score-stacking framework for Vietnamese student feedback sentiment analysis. Externally served LLMs were used as continuous sentiment scorers, and their outputs were calibrated by a lightweight local decision layer trained on labelled data.

At a high level, the framework consists of three stages. First, Vietnamese student feedback sentences were prepared from UIT-VSFC, with six labelled training examples removed for use as prompt calibration anchors. Second, three LLMs accessed through AIaaS inference platforms assigned each sentence a continuous sentiment score in [−1, 1]. Third, local decision models were trained on score-derived features to map these scores to negative, neutral, and positive labels. As illustrated in Fig. 1, the proposed framework separates externally served LLM-based continuous scoring from lightweight local decision learning.

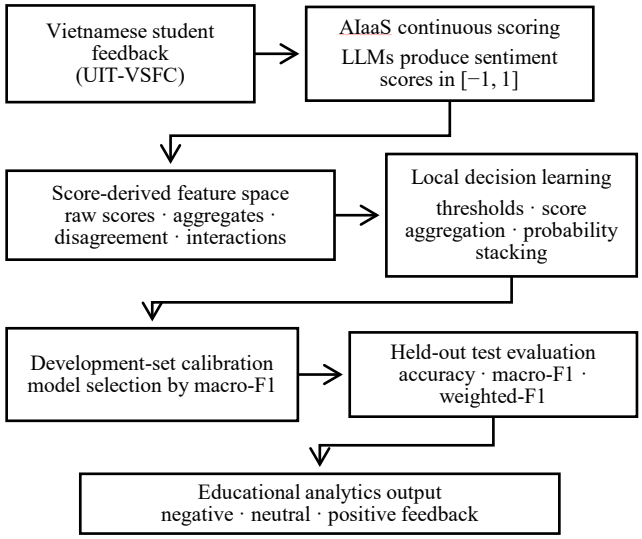

Fig. 1. Overview of the proposed AI-as-a-Service score-stacking framework for Vietnamese student feedback sentiment analysis.

Operationally, the framework comprises seven modules: dataset preparation and anchor exclusion, prompt construction, AIaaS scoring, response validation and score clipping, score storage and merging, score-derived feature construction, and local decision calibration, model selection, and held-out evaluation.

The design separates externally served semantic interpretation from local decision calibration, so the trainable component remains limited to low-dimensional score-derived features and lightweight classifiers.

The experiment followed the official

train-development-test logic of UIT-VSFC: training for fitting local meta-classifiers, development for threshold tuning, probability-bias adjustment, and model selection by macro-F1, and test for final reporting.

Algorithm 1. AIaaS score-stacking pipeline

Input: Prepared UIT-VSFC records with identifiers, texts, gold labels, and official train-development-test split; six calibration anchors; three AIaaS LLM scorers.

Output: Final sentiment predictions and evaluation metrics on the held-out test split.

1. Remove the six calibration anchors from the scoring, training, development, and test data.
2. For each LLM scorer, construct batched prompts containing the calibration anchors and feedback records with identifiers and sentence texts.
3. Call each externally served LLM with deterministic scoring settings where supported.
4. Parse the returned JSON object, validate each item, clip scores to the interval [−1, 1], and check that returned identifiers match input identifiers.
5. Save validated score outputs and resume scoring from existing identifiers when execution is interrupted.
6. Merge the three LLM score files with the gold labels and official split information.
7. Construct raw-score, aggregate, disagreement, and interaction feature from the three continuous LLM scores.
8. Fit local decision models on the training split.
9. Tune thresholds, score-aggregation weights, and probability-bias terms on the development split using macro-F1, and perform model selection on the same split.
10. Select the final model by development-set macro-F1 and report its performance once on the held-out test split.

C. Dataset and Preprocessing

The study used the UIT-VSFC dataset, a Vietnamese student feedback corpus annotated for sentiment classification. Each sentence is associated with one of three sentiment labels: negative, neutral, or positive. The original label encoding was preserved, with 0 representing negative, 1 representing neutral, and 2 representing positive. The dataset was prepared into train, development, and test splits. Six labelled examples from the training split were removed and used as few-shot calibration anchors in the LLM prompt. The calibration anchors used in the AIaaS scoring prompt are summarized in Table A1. These anchors were selected to represent strong positive, strong negative, and neutral feedback cases, while preserving their original gold labels. The selection criterion was prototypicality: anchors were chosen as clear examples of the original sentiment categories so that they could calibrate the continuous scoring scale without introducing additional topic or label definitions.

After removing the six calibration anchors, the prepared dataset contained 16,169 feedback sentences. The class distribution was imbalanced, with the neutral class substantially smaller than the negative and positive classes. This imbalance is central to the evaluation design because a model can obtain high accuracy by performing well on the two majority classes while still performing poorly on neutral comments. Table 1 reports the final dataset distribution used in the experiments.

Text preprocessing was intentionally minimal. The prepared sentences were not transformed into bag-of-words, Term Frequency-Inverse Document Frequency (TF-IDF), word embeddings, or other local lexical representations. Each sentence was sent as natural-language input to the LLM scoring layer, while the local model used only the resulting continuous scores and score-derived features.

The six removed calibration anchors were used only in the prompt and were excluded from model training and evaluation. They were intended to calibrate the interpretation of the [−1, 1] sentiment scale rather than to impose a closed set of output values.

Table 1. Dataset distribution after removing calibration anchors

Split	Negative	Neutral	Positive	Total
Train	5323	456	5641	11,420
Development	705	73	805	1583
Test	1409	167	1590	3166
Total	7437	696	8036	16,169

D. AI-as-a-Service Continuous Sentiment Scoring

The first modelling stage used three externally served large language models as continuous sentiment scorers. The models were Gemma-4-31B-IT accessed through Google AI Studio, Llama-3.1-8B-Instant accessed through Groq, and Qwen3-32B accessed through Groq. These names refer to the actual model families and model identifiers used in the experiment, whereas Google AI Studio and Groq functioned as inference platforms. This distinction is important because the study evaluates a multi-model AI-as-a-Service design rather than a multi-provider comparison.

Each model received the same task instruction and six calibration anchors. The prompt requested a continuous sentiment score in [−1, 1], allowed intermediate values for weak, mixed, or ambiguous sentiment, and instructed the models to preserve input identifiers and return machine-readable JavaScript Object Notation (JSON) without explanations or topic classification. At the module level, the input to each LLM scoring call consisted of the calibration anchors and a batch of feedback records containing sentence identifiers and sentence texts, while the expected output was a JSON object with an "items" array in which each item contained the original "id" and a continuous "score". The prompt also explicitly prohibited categorical sentiment labels, topic classification, explanations, markdown, and other additional text; the required machine-readable response format is specified in Algorithm 2. The complete prompt template used for continuous sentiment scoring is provided in Algorithm 3. The scorer-level configurations used for AIaaS continuous scoring are summarized in Table A2.

Formally, let x_i denote the i-th feedback sentence, and let $m \in \{\text{Gemma,Llama,Qwen}\}$ denote one of the three LLM scorers. Each model produces a continuous sentiment score:

$$s_i^{(m)} = f_m(x_i),\ s_i^{(m)} \in [-1,1] \quad (1)$$

The resulting three-dimensional raw score vector for sentence x_i is:

$$s_i = \left[s_i^{(\text{Gemma})}, s_i^{(\text{Llama})}, s_i^{(\text{Qwen})}\right] \quad (2)$$

Unlike direct LLM classification, this representation treats

model outputs as intermediate semantic signals. The final label is produced only after the three scores are passed to local calibration and stacking procedures. Continuous scores preserve sentiment intensity before thresholding while creating a compact representation for lightweight local modelling.

To improve reproducibility and reduce output variability, deterministic generation settings were used where supported. The scoring scripts implemented output validation, score clipping to [−1, 1], retry rules for malformed responses, and resume-safe writing to handle quota limits, rate limits, and service interruptions. A scoring batch was saved only after the parsed response passed item-level validation and the returned identifiers matched the input identifiers; for the Gemma scorer, mismatched or malformed batches in the Google AI Studio scoring script triggered recursive batch splitting, while Llama and Qwen call through Groq used bounded retry rules and stopped on persistent identifier mismatches.

The final scoring outputs contained one score from each of the three LLMs for every prepared sentence in the dataset. These score files were then merged with the gold labels and the official train, development, and test splits for local model training and evaluation. The AI-as-a-Service LLM scoring configuration is summarized in Table 2.

Table 2. AI-as-a-Service LLM scoring setup

Scorer	Inference Platform	Model ID	Role in the Framework	Output
Gemma	Google AI Studio	gemma-4-31b-it	Continuous sentiment scorer	Score in [−1, 1]
Llama	Groq	llama-3.1-8b-instant	Continuous sentiment scorer	Score in [−1, 1]
Qwen	Groq	qwen/qwen3-32b	Continuous sentiment scorer	Score in [−1, 1]

Note. Model IDs are reported exactly as specified in the scoring scripts.

For the LLM scoring layer, all scoring calls used temperature = 0 where supported. The Gemma scorer was accessed through Google AI Studio with an application/JSON response format and a structured response schema. The Llama and Qwen scorers were accessed through Groq with JSON-object response formatting and a maximum completion length of 4096 tokens. Top-p and seed values were not manually specified in the scoring scripts. Hyperparameters and tuning grids for the local decision layer are reported separately in the local modelling subsection. The inputs and outputs of each module in the AIaaS score-stacking pipeline are summarized in Table A3.

E. Score-Derived Feature Construction

After the three LLM scores were obtained, each feedback sentence was represented by a low-dimensional vector of score-derived features. These features allowed the local decision layer to use both individual model scores and inter-model relationships such as agreement, disagreement, and score dispersion. The rationale was that sentiment labels may depend not only on the average orientation of the scorers, but also on class-specific evidence: minimum and maximum scores capture strongly negative or positive signals, range and pairwise differences capture ambiguity or scorer disagreement, and interaction terms capture whether two scorers jointly support the same sentiment direction.

The feature construction procedure therefore included three types of variables. The first type consisted of the raw scores generated by the three LLMs: $s_i^{(\text{Gemma})}$, $s_i^{(\text{Llama})}$, $s_i^{(\text{Qwen})}$. The second type consisted of aggregate score statistics. These features summarize the central tendency and dispersion of the model scores:

$$\mu_i = \frac{1}{3}\left(s_i^{(\text{Gemma})} + s_i^{(\text{Llama})} + s_i^{(\text{Qwen})}\right) \tag{3}$$

$$s_{i,\min} = \min_m s_i^{(m)} \tag{4}$$

$$s_{i,\max} = \max_m s_i^{(m)} \tag{5}$$

$$r_i = s_{i,\max} - s_{i,\min} \tag{6}$$

Here, μ_i represents the average sentiment orientation, while r_i represents the degree of disagreement among the three scorers. A small range indicates that the models broadly agree, whereas a large range suggests divergent interpretations of the same feedback sentence.

The third type consisted of pairwise difference and interaction features. Pairwise differences capture directional disagreement between model scores:

$$d_i^{(a,b)} = s_i^{(a)} - s_i^{(b)} \tag{7}$$

where a and b are two different LLM scorers. Absolute differences were also used to represent the magnitude of disagreement regardless of direction:

$$\left|d_i^{(a,b)}\right| = \left|s_i^{(a)} - s_i^{(b)}\right| \tag{8}$$

Interaction features, such as score products, were included to capture whether the models jointly indicate the same sentiment direction. For example, if two scores have the same sign and large magnitude, their product is positive and large; if they have opposite signs, the product becomes negative. Such features allow the local decision layer to model relationships that are not captured by a simple average.

The final score-derived feature vector for sentence x_i can be written as:

$$z_i = \phi\left(s_i^{(\text{Gemma})}, s_i^{(\text{Llama})}, s_i^{(\text{Qwen})}\right) \tag{9}$$

where $\phi(\cdot)$ denotes the feature construction function that maps the three raw LLM scores into raw-score, aggregate, disagreement, and interaction features.

This lightweight feature design uses only the three continuous LLM scores and does not require token-level representations, hidden states, embeddings, or additional text encoders. It remains richer than naive soft voting because it preserves agreement, disagreement, and score-intensity information for the local decision layer.

F. Local Decision Layers and Baseline Models

The local modelling stage compared increasingly flexible ways of converting continuous LLM scores into sentiment

labels, from fixed thresholds and calibrated score aggregation to trainable probability stacking. The first group of baselines consisted of fixed-threshold single-model classifiers. For each LLM scorer, the continuous score was mapped to a sentiment label using predefined thresholds. Given a score $s_i^{(m)}$, the fixed-threshold decision rule was:

$$\hat{y}_i = \begin{cases} 0, & s_i^{(m)} \le t_{\text{neg}} \\ 2, & s_i^{(m)} \ge t_{\text{pos}} \\ 1, & t_{\text{neg}} < s_i^{(m)} < t_{\text{pos}} \end{cases} \quad (10)$$

where 0, 1, and 2 denote negative, neutral, and positive sentiment, respectively. This baseline represents the most direct use of an LLM score: each model independently produces a continuous score, and the score is converted into a label without learning a local classifier.

The second group consisted of tuned-threshold single-model classifiers. These models used the same decision rule but selected t_{neg} and t_{pos} on the development set to maximize macro-F1. This baseline tests whether simple score calibration is sufficient to adapt each LLM scorer to the target dataset. It also serves as an important comparison against more complex stacking models, because a strong tuned single-model baseline may reduce the apparent need for ensemble learning.

The third group consisted of score-level aggregation methods. Equal soft aggregation used the unweighted average of model scores. For the three-model setting, the aggregated score was:

$$\bar{s}_i = \frac{s_i^{(\text{Gemma})} + s_i^{(\text{Llama})} + s_i^{(\text{Qwen})}}{3} \quad (11)$$

Weighted score aggregation generalized this rule by learning non-negative model weights on the development set:

$$\bar{s}_i = w_{\text{Gemma}} s_i^{(\text{Gemma})} + w_{\text{Llama}} s_i^{(\text{Llama})} + w_{\text{Qwen}} s_i^{(\text{Qwen})} \quad (12)$$

subject to:

$$w_{\text{Gemma}} + w_{\text{Llama}} + w_{\text{Qwen}} = 1 \quad (13)$$

The aggregated score was then converted into a sentiment label using either fixed or tuned thresholds. These models represent linear score-level ensemble baselines. They test whether the main benefit of multiple LLM scorers can be captured by a simple weighted average.

The fourth group consisted of trainable probability-stacking models. In this setting, the score-derived feature vector z_i was used as input to a local classifier:

$$p_i = g(z_i) \quad (14)$$

where $p_i = [p_{i,0}, p_{i,1}, p_{i,2}]$ denotes the predicted probability distribution over negative, neutral, and positive classes. The final label was selected by:

$$\hat{y}_i = \underset{c \in \{0,1,2\}}{\arg\max}\ p_{i,c} \quad (15)$$

Candidate meta-learners included logistic regression, random forest, ExtraTrees, histogram gradient boosting, support vector classification, and multilayer perceptron. These lightweight models can learn from low-dimensional tabular features without text representations or LLM fine-tuning.

A probability-bias tuning variant was also evaluated for stacking models. Instead of directly selecting the class with the highest predicted probability, the decision rule adjusted the class probabilities by development-set-tuned class-specific bias terms:

$$\hat{y}_i = \underset{c \in \{0,1,2\}}{\arg\max} \left(p_{i,c} + b_c\right) \quad (16)$$

where b_c is the bias term for class c. This adjustment allows the final decision rule to compensate for class imbalance and improve macro-F1 on the development set. Importantly, these bias terms were tuned only on the development set and were not optimized on the test set.

Together, these model groups provide an ablation design for testing whether trainable score stacking adds value beyond direct scoring, threshold calibration, and score aggregation. The baseline set was therefore designed to compare decision strategies within the proposed score-based AIaaS framework, rather than to exhaust all possible text-classification alternatives such as direct categorical LLM prompting, TF-IDF classifiers, embedding-based classifiers, or fine-tuned transformer models. The experimental model groups evaluated in this study are summarized in Table 3.

Table 3. Experimental model groups

Model group	Input	Trainable Local Component	Purpose
Single-model fixed threshold	One LLM score	No	Direct score-to-label baseline
Single-model tuned threshold	One LLM score	Thresholds tuned on development set	Tests simple score calibration
Equal soft aggregation	Mean of two or three LLM scores	No	Tests naive score averaging
Weighted score aggregation	Weighted LLM score average	Weights and thresholds tuned on development set	Tests linear score-level ensemble
Probability stacking	Score-derived feature vector	Local classifier and optional probability-bias tuning	Tests trainable score stacking

To make the local modelling procedure reproducible, the main hyperparameters and development-set tuning grids used for these model groups are reported in Table 4.

G. Evaluation Protocol and Model Selection

Because the dataset is imbalanced and the neutral class is substantially underrepresented, macro-F1 was used as the primary model selection metric. Accuracy and weighted-F1 were reported as complementary metrics.

For each model, class-level precision, recall, and F1-score were computed for negative, neutral, and positive sentiment. Overall accuracy was computed as the proportion of correctly classified instances:

$$\text{Accuracy} = \frac{\sum_{c=0}^{2} TP_c}{N} \quad (17)$$

where TP_c denotes the number of true positives for class (c),

and (N) denotes the total number of evaluated instances.

Table 4. Local decision-layer hyperparameters and tuning grids

Component	Settings used in the local decision layer
Fixed-threshold baselines	Negative threshold = −0.33; positive threshold = 0.33
Tuned single-score thresholds	Negative and positive thresholds searched on the development set from −0.95 to 0.95 with a step size of 0.01, subject to negative threshold < positive threshold
Three-model weighted score aggregation	Non-negative Gemma, Llama, and Qwen weights constrained to sum to 1; coarse search used weight step = 0.10 and threshold step = 0.05; refined search used weight step = 0.02 and threshold step = 0.01 around the coarse optimum
Probability-bias tuning	Negative bias fixed at 0 for identifiability; neutral bias searched from −0.30 to 0.50 with a step size of 0.01; positive bias searched from −0.30 to 0.30 with a step size of 0.01
Logistic regression stacking	StandardScaler; lbfgs solver; max_iter = 2000; class_weight = balanced; random_state = 42
Random forest stacking	n_estimators = 500; max_depth = 6; min_samples_leaf = 3; class_weight = balanced; random_state = 42
ExtraTrees stacking	n_estimators = 500; max_depth = 6; min_samples_leaf = 3; class_weight = balanced; random_state = 42
Histogram gradient boosting stacking	max_iter = 300; learning_rate = 0.04; max_leaf_nodes = 15; l2_regularization = 0.05; random_state = 42
SVC-RBF stacking	StandardScaler; kernel = rbf; C = 3.0; gamma = scale; probability = true; class_weight = balanced; random_state = 42
MLP stacking	Maximum epochs = 300; patience = 30; batch size = 64; Adam learning rate = 1×10^{-3}; weight decay = 1×10^{-4} in the PyTorch implementation; class-weighted loss used to mitigate imbalance

For each class (c), precision, recall, and F1-score were defined as:

$$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c} \tag{18}$$

$$\text{Recall}_c = \frac{TP_c}{TP_c + FN_c} \tag{19}$$

$$F1_c = \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \tag{20}$$

Macro-F1 was then computed as the unweighted average of the three class-specific F1-scores:

$$\text{Macro-F1} = \frac{1}{3}\sum_{c=0}^{2} F1_c \tag{21}$$

Weighted-F1 was computed as the support-weighted average:

$$\text{Weighted-F1} = \sum_{c=0}^{2} \frac{n_c}{N} F1_c \tag{22}$$

where n_c is the number of instances belonging to class c. Macro-F1 was prioritized because it gives equal importance to negative, neutral, and positive classes, whereas weighted-F1 is more strongly influenced by the majority classes.

The training split was used to fit local stacking classifiers, the development split was used to tune thresholds, score-aggregation weights, and probability-bias terms and to select the final model, and the test split was reserved for final reporting. The final selected model was the one with the highest development-set macro-F1. To assess the uncertainty of the held-out test results, percentile bootstrap analysis was conducted by resampling test instances with replacement 10,000 times and recalculating macro-F1 for each resample. Paired bootstrap comparisons were also conducted between the final selected model and the main competing methods by recalculating macro-F1 differences on the same bootstrap resamples. These bootstrap analyses were used only for post-selection interpretation and did not affect model selection.

Ablation analysis across model groups was used to assess whether performance gains came from single-model calibration, score averaging, weighted aggregation, or trainable probability stacking. For reproducibility, the analysis pipeline exported merged predictions, the final metrics summary, the development-based model selection record, per-method classification reports, confusion matrices, the ablation summary, feature-importance outputs, Multilayer Perceptron (MLP) training curves, and auxiliary Qwen contribution cases. Appendix A reports the scoring prompt template, the complete calibration-anchor table, scorer-level configuration, validation logic, scorer-level output files, and main analysis scripts, thereby making the AIaaS scoring and local decision pipeline auditable.

IV. Results

A. Overall Model Performance

The experimental results suggest that trainable score stacking was associated with the strongest macro-F1 performance among the methods evaluated in this study. Fig. 2 presents the test macro-F1 scores of representative single-model baselines, score aggregation methods, and probability-stacking models.

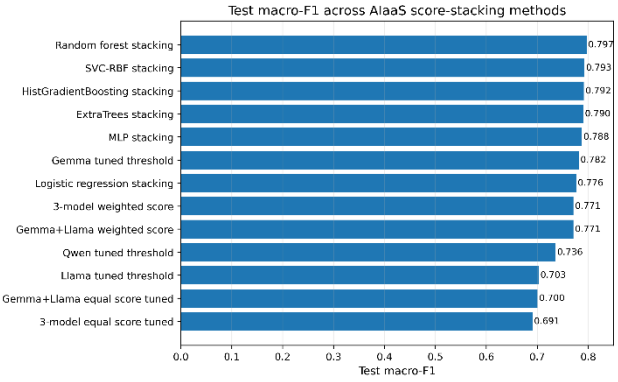

Fig. 2. Test macro-F1 across representative AI-as-a-Service score-stacking methods.

Table 5 reports the development and test performance of the main evaluated methods, organized by modelling stage.

Because the differences among the probability-stacking models were small, the results are interpreted primarily as evidence for the usefulness of the score-derived representation and development-based local calibration rather than as evidence that one specific meta-classifier is intrinsically superior. Bootstrap analysis on the held-out test split provided further support for this cautious interpretation. The final selected ExtraTrees model obtained a test macro-F1 of 0.790, with a 95% percentile bootstrap confidence interval of [0.766, 0.813]. Paired bootstrap comparisons, calculated as

the macro-F1 difference between ExtraTrees and each comparator, showed that the 95% confidence intervals included zero for the main competing methods, including random forest [−0.019, 0.006], histogram gradient boosting [−0.018, 0.016], Support Vector Classifier-Radial Basis Function (SVC-RBF) [−0.018, 0.014], MLP [−0.011, 0.017], logistic regression [−0.001, 0.028], three-scorer weighted score aggregation [−0.001, 0.039], and Gemma tuned thresholding [−0.010, 0.028]. These results indicate that the test-set differences among the strongest methods should not be overinterpreted as statistically decisive model rankings.

Table 5. Development and test performance of the main evaluated methods

Group	Method	Dev (Acc./Macro-F1/W-F1)	Test (Acc./Macro-F1/W-F1)
Fixed single-score baselines	Gemma fixed threshold	0.799 / 0.658 / 0.839	0.798 / 0.658 / 0.833
	Llama fixed threshold	0.841 / 0.696 / 0.839	0.837 / 0.692 / 0.836
	Qwen fixed threshold	0.795 / 0.657 / 0.831	0.793 / 0.667 / 0.826
Tuned single-score baselines	Gemma tuned threshold	0.925 / 0.800 / 0.922	0.913 / 0.782 / 0.909
	Llama tuned threshold	0.850 / 0.725 / 0.850	0.845 / 0.703 / 0.844
	Qwen tuned threshold	0.868 / 0.740 / 0.875	0.864 / 0.736 / 0.871
Three-scorer score aggregation	3-scorer equal score fixed	0.832 / 0.686 / 0.855	0.825 / 0.682 / 0.848
	3-scorer equal score tuned	0.855 / 0.707 / 0.863	0.851 / 0.691 / 0.858
	3-scorer weighted score tuned	0.925 / 0.801 / 0.923	0.911 / 0.771 / 0.907
Probability stacking	Logistic regression + prob. bias	0.922 / 0.802 / 0.929	0.908 / 0.776 / 0.915
	Random forest + prob. bias	0.939 / 0.837 / 0.939	0.919 / 0.797 / 0.920
	ExtraTrees + prob. bias †	0.942 / 0.843 / 0.943	0.921 / 0.790 / 0.921
	HistGradientBoosting + prob. bias	0.946 / 0.841 / 0.945	0.931 / 0.792 / 0.927
	SVC-RBF + prob. bias	0.939 / 0.826 / 0.940	0.925 / 0.793 / 0.925
	MLP + prob. bias	0.936 / 0.826 / 0.938	0.919 / 0.788 / 0.920

Note. W-F1 = weighted-F1; prob. bias = probability-bias tuning. † indicates the final model selected according to development-set Macro-F1. Two-scorer exploratory aggregation ablations are not shown in the main table because the primary framework uses three LLM scorers.

The best-performing group consisted of probability-stacking methods trained on score-derived features. Random forest stacking achieved the highest observed test macro-F1 of 0.797, followed by SVC-RBF stacking with 0.793, histogram gradient boosting stacking with 0.792, ExtraTrees stacking with 0.790, and MLP stacking with 0.788. These results suggest that the multi-model score space can provide useful input for a lightweight local decision layer.

Among the single-model baselines, Gemma with tuned thresholds was the strongest, reaching a test macro-F1 of 0.782. This suggests that a single AIaaS LLM can provide a useful sentiment signal when its continuous scores are calibrated on development data. However, the best stacking models still improved over this baseline, suggesting that the score-derived feature space contains useful information beyond the output of the strongest individual scorer.

The score aggregation baselines were weaker. The three-model weighted score aggregation achieved a test macro-F1 of 0.771. Equal score averaging achieved lower macro-F1, with the three-model equal-score tuned method reaching only 0.691. These results suggest that multiple LLM scores provided more useful input in this evaluation when interpreted by a trainable local decision model rather than averaged directly.

B. Ablation Analysis: From Direct Scoring to Probability Stacking

The ablation results clarify where the performance gains came from. The first comparison concerns direct single-model scoring. Threshold tuning improved the usability of LLM scores, particularly for Gemma, supporting the interpretation that raw continuous scores should not be treated as directly calibrated class decisions.

The second comparison concerns naive multi-model aggregation. Equal averaging achieved lower macro-F1 than tuned single-model scoring, suggesting that adding more LLMs can dilute useful model-specific signals when their scores are simply averaged.

The third comparison concerns weighted score aggregation. Weighted models improved over equal averaging but remained limited, suggesting that linear score combination may lose information about disagreement, score range, and class-specific reliability.

The fourth comparison concerns probability stacking. The highest macro-F1 values in the ablation were observed when score-derived features were passed to local classifiers such as random forest, ExtraTrees, histogram gradient boosting, SVC-RBF, and MLP. Because several stacking models performed similarly well, the gain appears to reflect the usefulness of the score-derived representation rather than a single classifier architecture.

Although random forest stacking produced the highest observed test macro-F1, the final model was selected according to development-set macro-F1 rather than test-set performance. Under this protocol, ExtraTrees stacking with probability-bias tuning was retained as the final model. As reported in Table 5, ExtraTrees achieved the highest development-set macro-F1 among the evaluated methods, whereas random forest had the highest observed test macro-F1 but was not selected using test information. This model achieved a test macro-F1 of 0.790, a test accuracy of approximately 0.921, and a test weighted-F1 of approximately 0.921. This preserves the separation between model selection and final test reporting.

C. Comparison with Prior UIT-VSFC Baselines

To contextualize the results, Table 6 compares the proposed AI-as-a-Service score-stacking framework with representative prior studies on UIT-VSFC. This comparison is intended as contextual positioning rather than a strict head-to-head benchmark, because the cited studies differ in preprocessing, data splits, feature representations, training protocols, and metric reporting. Its purpose is not to claim a strict leaderboard position, but to situate the proposed framework within Vietnamese student feedback sentiment analysis.

Table 6. Contextual comparison with representative UIT-VSFC baselines

Study	Approach	Reported Setting	Reported Result	Comparability Note
[5]	Naive Bayes / Maximum Entropy with *n*-gram features	Original UIT-VSFC baseline; official train/dev/test split	MaxEnt sentiment F1 = 87.94%; neutral F1 = 33.99%	Strong dataset reference; metric is overall/average F1, not directly equivalent to macro-F1
[6]	LSTM, Dependency Tree-LSTM, and LD-SVM	Deep learning and hybrid recurrent models on UIT-VSFC	Best reported F1 = 90.2%; accuracy = 90.7%	Strong neural baseline; reported metric differs from this study's primary macro-F1
[19]	NB, MaxEnt, LSTM, Bi-LSTM with Word2Vec	80:20 split; micro-average evaluation	Bi-LSTM F1 = 92.0%	Uses a different split and micro-averaging; not directly comparable to development-selected macro-F1
[20]	Text-CNN and XLM-R with Vietnamese emotion lexicon	UIT-VSFC evaluation with macro-F1 and weighted-F1	XLM-R + VnEmoLex macro-F1 = 83.40%; weighted-F1 = 93.97%	Strong transformer/lexicon baseline; higher macro-F1 than the present framework
Present study	AIaaS continuous LLM scoring + local probability stacking	Official train/dev/test after removing six calibration anchors; selected by development-set macro-F1	ExtraTrees stacking test macro-F1 = 79.04%; accuracy = 92.07%; weighted-F1 = 92.11%	No LLM fine-tuning or local text encoder training; local model uses only score-derived features

Note. The reported metrics are presented as stated in the original studies and should not be interpreted as directly interchangeable. In particular, overall F1, micro-F1, macro-F1, and weighted-F1 differ in how they aggregate class-level performance, especially under neutral-class imbalance.

At a contextual level, the proposed framework is broadly comparable to several traditional and recurrent baselines, despite not training a full text classifier or fine-tuning a language model.

The framework does not surpass the strongest transformer-based baseline, particularly XLM-R combined with a Vietnamese emotion lexicon. Its contribution is therefore not a state-of-the-art claim on UIT-VSFC, but a service-oriented deployment architecture based on external LLM scoring and lightweight local decision learning.

D. Class-level Performance and Neutral-class Difficulty

The class-level results reveal that the main challenge of the task lies in the neutral class. Across the strongest models, F1-scores for negative and positive feedback were consistently high, typically above 0.93. By contrast, neutral F1 remained substantially lower, around 0.48–0.50 for the strongest methods. This gap reflects both the class imbalance of the dataset and the semantic ambiguity of neutral feedback.

The detailed class-level prediction patterns are illustrated in Fig. 3. The selected ExtraTrees model correctly classified 1293 of 1409 negative test instances, corresponding to a row-level recall of 91.8%. It also correctly classified 1540 of 1590 positive instances, corresponding to a row-level recall of 96.9%. These results show that the model achieved high recall for the two majority sentiment classes. The neutral class was more difficult: 82 of 167 neutral instances were correctly classified, corresponding to a recall of 49.1%. Among the misclassified neutral instances, 28 were predicted as negative and 57 were predicted as positive. In row-normalized terms, 16.8% of neutral instances were predicted as negative and 34.1% as positive, confirming that neutral comments were frequently absorbed into the two affectively clearer classes. This pattern suggests that neutral feedback was more often absorbed into the positive class than into the negative class.

Neutral feedback therefore appears to function as a boundary region in educational sentiment analysis, often containing descriptive, mixed, suggestive, or weakly affective expressions that overlap with positive and negative classes. Inspection of the misclassified neutral instances in the held-out test split showed two recurrent error patterns. Some neutral comments were predicted as positive when they contained positive lexical cues or polite evaluative wording, such as "ổn" (acceptable/okay), "bài tập đầy đủ" (sufficient exercises), or "ví dụ phù hợp với nội dung kiến thức, hướng dẫn chi tiết" (examples suitable for the lesson content, with detailed guidance). Conversely, some neutral comments were predicted as negative when they described workload, time constraints, absence, or difficulty, such as "nhiều bài thuyết trình" (many presentations), "quá nhiều kiến thức cần nắm nhưng thời gian có hạn" (too much knowledge to cover in limited time), or "thầy có vẻ hơi bận rộn, gặp thầy có vẻ khó" (the teacher seemed busy and difficult to meet). These cases show that many neutral labels in UIT-VSFC are not affectively empty; rather, they often encode weak evaluation, contextual description, or implicit suggestion.

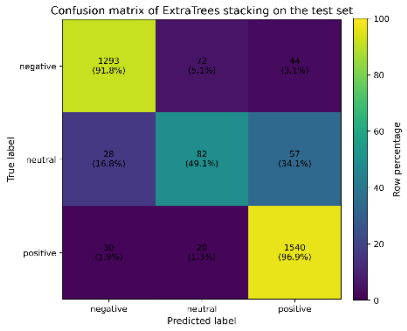

Fig. 3. Confusion matrix of the final selected ExtraTrees stacking model on the test set.

This class-level imbalance also explains why macro-F1 is more informative than accuracy or weighted-F1 for this task: neutral-class errors would otherwise be masked by strong performance on the two majority classes. Fig. 4 presents the class-level F1-scores for the selected single-model and stacking methods, highlighting the consistently lower performance on the neutral class.

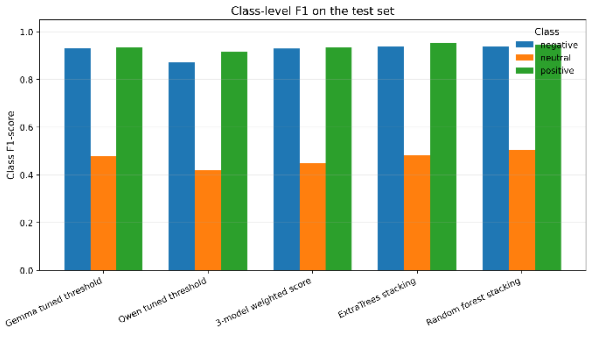

Fig. 4. Class-level F1-scores for selected baselines and stacking models on the test set. Negative and positive classes were consistently easier than the neutral class, supporting the use of macro-F1 as the primary evaluation metric.

E. Feature Importance and Interpretation of the Stacking Model

Feature importance analysis provides insight into how the selected ExtraTrees stacking model used the score-derived feature space. The most important feature was the Gemma score, suggesting that the Gemma score carried the strongest individual predictive signal among the three LLM scorers. However, the model did not rely exclusively on this single score. Several aggregate and multi-model features also appeared among the most important predictors, including the three-model minimum, the Gemma-Llama minimum, the Qwen score, the three-model mean, and the Gemma-Llama mean. Fig. 5 presents the most important score-derived features identified by the final selected ExtraTrees stacking model.

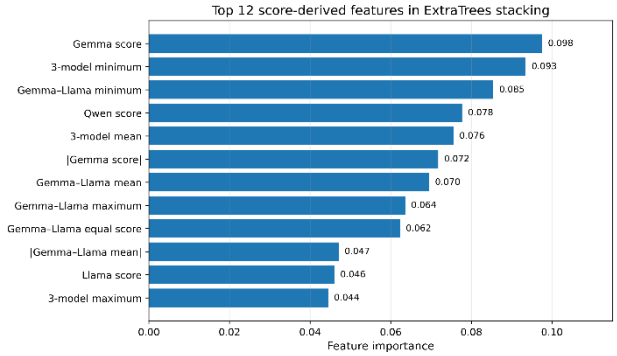

Fig. 5. Top score-derived features in the selected ExtraTrees stacking model.

This pattern indicates that the stacking model did not merely reproduce the strongest single LLM. It also used aggregate and cross-model features, including the three-model minimum, three-model mean, and Qwen score. Minimum-based, mean-based, and absolute-value features are interpretable because they respectively capture negative or ambiguous signals, overall sentiment direction, and sentiment strength.

V. Discussion

A. From Model Performance to Deployment Architecture

The results should be interpreted not only as a comparison of sentiment classification methods but also as evidence supporting a deployment architecture for educational analytics. While prior student feedback studies have emphasized supervised NLP performance, the present study addresses a different institutional question: how a university with limited AI infrastructure can operationalize feedback analytics without training, fine-tuning, or hosting LLMs locally.

The proposed framework addresses this question by separating externally served semantic interpretation from local decision calibration. It shifts the institutional burden from building large models to orchestrating external model services and calibrating their outputs for a local analytic task.

This distinction is important in the context of higher education AI adoption. Recent work on generative AI in higher education has emphasized that the challenges of adoption are not limited to model capability. They also include technological readiness, data privacy, equity, transparency, stakeholder roles, and institutional governance [10, 11]. Similarly, institutional LLM platform studies have shown that universities increasingly need mechanisms for secure inference, access control, cost management, API integration, and responsible experimentation [7, 8]. The present study contributes to this line of thinking by demonstrating a task-specific implementation pattern: external LLMs can be treated as semantic services, while the institution retains a local and auditable decision layer.

The empirical results support this architecture while also placing boundaries around it. The framework was competitive with several traditional and recurrent UIT-VSFC baselines, but did not surpass the strongest transformer-and-lexicon baseline. Its value therefore lies not in state-of-the-art performance, but in a feasible and modular pathway for institutions unable to maintain full supervised NLP pipelines or fine-tuned transformer models.

The contribution is therefore architectural and implementation-oriented rather than solely algorithmic. The study shows that a higher education institution can externalize the computationally intensive language understanding layer while internalizing task-specific calibration. This aligns with the broader logic of educational data mining and learning analytics, where predictive models are used to transform educational data into evidence for monitoring, quality improvement, and decision-making [1].

B. Why Score Stacking Matters more than Naive Multi-model Averaging

The ablation analysis shows that the value of multi-model AIaaS does not come from using more models mechanically. Equal averaging was weaker than the strongest tuned single-model baseline, and weighted aggregation produced only limited gains, indicating that useful information may be lost when the combination rule is too rigid. Averaging-based methods impose a single global combination rule, whereas probability stacking can learn class-specific decision boundaries from raw scores, score ranges, inter-model disagreement, and interaction features. This interpretation is consistent with stacked generalization theory, in which a higher-level learner is trained to model the behavior and errors of lower-level predictors, and with calibration-oriented classification, where confidence-like outputs become more useful when they are adjusted against held-out development data [15, 16].

The results support this design. Probability-stacking models formed the strongest group, suggesting that disagreement, score ranges, minimum and maximum values, and interaction patterns carry information that is lost when scores are compressed into a single average. Feature importance analysis reinforces this interpretation because the selected ExtraTrees model used both individual and aggregate score-derived features.

This has an important implication for multi-LLM research: the value of an additional LLM may appear not through standalone accuracy or simple voting, but through the structure it adds to the score space. In this study, Qwen was not the strongest single scorer and three-model averaging was not optimal, yet Qwen-related features still contributed to the selected stacking model.

For educational analytics, the implication is that institutions need a principled mechanism for calibrating,

comparing, and combining external AI outputs. Score stacking provides such a mechanism by turning multiple LLM scores into a local, trainable, and evaluable decision process.

C. Continuous Sentiment Scores and the Problem of Neutral Feedback

The findings support representing LLM outputs as continuous sentiment scores before final classification. This is consistent with sentiment analysis literature that treats polarity not only as discrete categories, but also as intensity along a continuous or ordinal scale [13, 14]. For student feedback, this intermediate representation is useful because comments are often short, mixed, descriptive, or weakly affective.

The neutral class illustrates this problem. Negative and positive feedback were classified with high class-level F1-scores and recall, whereas neutral feedback remained substantially more difficult to classify, consistent with prior UIT-VSFC studies that link neutral sentiment to low frequency and semantic ambiguity [5, 6]. Neutral comments may include factual descriptions, mild suggestions, mixed evaluations, or context-dependent statements that lie near the boundary between positive and negative interpretation. The test-set error analysis in this study supports this interpretation: neutral comments containing weakly positive cues tended to be absorbed into the positive class, whereas comments describing workload, time constraints, absence, or difficulty tended to be absorbed into the negative class.

The relatively low neutral F1 has both technical and institutional implications. Technically, it reflects class imbalance and decision-boundary difficulty. Institutionally, it shows that neutral comments may require careful treatment because descriptive or weakly evaluative feedback can still contain actionable information for quality assurance.

This reinforces the importance of macro-F1, class-level evaluation, and explicit calibration. In this study, macro-F1 is aligned with the educational purpose of keeping all three sentiment classes visible, especially the underrepresented neutral class.

The difficulty of neutral feedback also suggests that future work should move beyond three-class sentiment classification. Neutral comments may be descriptive, mixed, uncertain, procedural, or implicitly evaluative. Future systems should therefore combine sentiment scoring with uncertainty estimation, aspect extraction, suggestion detection, or human review for borderline cases.

D. Implications for Low-infrastructure Higher Education Analytics

Because this study evaluates a computational architecture rather than an implemented institutional intervention, the implications discussed below should be read as design and governance implications rather than as direct evidence of managerial effectiveness.

The findings have practical implications for higher education institutions that possess educational data but lack extensive AI infrastructure. In such contexts, the key challenge is not only data availability, but the ability to convert data into reliable evidence without building a full AI engineering stack. Operational deployment would still require routine logging of API calls, latency, retry and failure rates, rate-limit events, and cost per 1000 processed feedback records, because these factors determine whether an AIaaS design remains sustainable in a real feedback-analytics workflow. As an illustrative list-price estimate based on the public pricing pages available, the two paid Groq-hosted scorers used in this study would cost approximately US$0.69 for 16,169 feedback records, or about US$0.043 per 1000 records, excluding retry overhead [30]; the Google AI Studio Gemma scorer had no listed marginal API charge in the free-tier setting but would still be subject to quota and rate-limit constraints [31] (although the documentation portal is branded as Gemini API, it covers multiple model families served through Google AI Studio, including Gemma).

The proposed framework suggests a layered pathway between building local AI models and outsourcing the entire analytic process. External AIaaS models provide language understanding capabilities, while the institution retains a local layer for calibration, evaluation, and task-specific adaptation. External LLMs are therefore not treated as final authorities, but as sources of intermediate signals interpreted by a locally trained and evaluated model.

This has practical value for educational management because feedback analytics must support institutional reflection and action, not merely prediction. Course dashboards, teaching quality monitoring, curriculum improvement, and student experience analysis require transparent evaluation and class-sensitive interpretation. The lightweight local layer allows institutions to adapt thresholds and model selection to priorities such as neutral or ambiguous feedback.

The framework also supports modularity. Because LLMs are used as external scoring services, one model can be replaced or added without redesigning the entire system. If pricing, rate limits, privacy conditions, or model availability change, the institution can update the scoring layer while preserving local feature construction, decision learning, and development-test evaluation.

At the same time, the framework does not eliminate governance responsibilities. AI-as-a-Service introduces dependencies on external providers, including data privacy, cost predictability, service continuity, model versioning, and reproducibility. Universities need rules on what educational data may be sent to external services, how identifiers are protected, how outputs are logged, how model changes are documented, and how analytic reliability is evaluated for decision support.

A further implication concerns the meaning of AI capacity in higher education. AI capacity should not be understood only as the ability to train or host large models locally. For many institutions, it may instead involve task design, service selection, output validation, decision calibration, performance monitoring, and educational interpretation. In this sense, AI capacity is less about owning computational infrastructure than governing an analytic workflow.

From a digital transformation perspective, successful AI adoption in higher education should be assessed not only by model sophistication, but also by whether the AI workflow is feasible, auditable, adaptable, and aligned with institutional decision-making. For low-infrastructure institutions, the strategic question is not simply how to acquire AI, but how to orchestrate AI services into reliable, locally meaningful, and

governable analytic systems.

E. Limitations and Future Research

This study has several limitations. First, the empirical evaluation was conducted on a single benchmark dataset, UIT-VSFC. Although this dataset is highly relevant to Vietnamese student feedback sentiment analysis, comments from other universities, disciplines, regions, and institutional cultures may differ in vocabulary, tone, implicit meaning, and sentiment distribution. Future research should test the framework on additional Vietnamese educational feedback datasets and institutional course evaluation data.

Second, the study evaluated three externally served LLMs selected under feasible access, cost, and quota conditions. The results should therefore be interpreted as evidence for the proposed architecture rather than as a definitive comparison of all possible LLMs or inference providers. Future studies should examine additional model families, larger or domain-specific models, multilingual models, locally hosted open models, and provider-level effects such as structured output support, latency, rate limits, cost, privacy and data-retention terms, provider dependence, stability, and model versioning. A particularly important extension would be to repeat the same scoring prompts across multiple execution dates and provider settings to estimate the temporal stability of externally served LLM scores. The present study did not empirically evaluate direct categorical prompting of the type examined in recent Vietnamese LLM studies [24, 25], nor did it evaluate TF-IDF classifiers, embedding-based classifiers, or fine-tuned Vietnamese and multilingual transformer models under the same experimental protocol. Consequently, the reported results do not establish superiority over these alternative model families. Future work should compare them under the same data split, model-selection rule, and reporting protocol. Additional comparisons should examine whether post-hoc calibration methods such as temperature scaling or isotonic regression, as well as pseudo-labeling or weak-supervision strategies, can improve neutral-class reliability under the same evaluation protocol. For the neutral class specifically, future work should examine neutral-focused data augmentation, class-balanced development objectives, ordinal or uncertainty-aware decision rules, and active-learning review of boundary cases. Given the limited neutral-class performance, the proposed framework should not be used as a fully automated decision system for high-stakes educational actions. Neutral, ambiguous, or low-confidence comments should be flagged for human review. Exact wall-clock processing time and the invalid-response rate were not retained as independent runtime artifacts. Although the scoring pipeline recorded validated outputs and supported retry, batch splitting, and resumed execution, these two operational quantities could not be reported retrospectively and therefore constitute a methodological limitation of the present study.

Third, the local decision layer used only score-derived features. This was intentional because the study aimed to test a lightweight, low-infrastructure design, but it also limits the information available to the meta-classifier. Future work could combine AI-as-a-Service sentiment scores with local features such as topic categories, aspect tags, course-level variables, instructor-level variables, or human-coded uncertainty labels. Such extensions may improve neutral-class performance and support richer tasks such as aspect-based sentiment analysis, suggestion mining, urgency detection, and feedback summarization.

Finally, this study did not evaluate deployment outcomes in a real institutional workflow. It demonstrated technical feasibility and predictive performance, but did not examine how instructors, managers, or quality assurance staff would interpret and use the outputs. Future implementation studies should investigate dashboard design, user trust, decision impact, privacy governance, cost sustainability, and integration with existing quality assurance processes.

VI. Conclusion

This study proposed and evaluated an AI-as-a-Service score-stacking framework for Vietnamese student feedback sentiment analysis. Externally served LLMs were used as continuous sentiment scorers, and their scores were transformed into a local feature space for lightweight decision learning over negative, neutral, and positive labels.

The empirical results on UIT-VSFC suggest that, in this experimental setting, probability-stacking models achieved higher macro-F1 than direct fixed-threshold baselines, tuned single-model baselines, equal score averaging, and weighted score aggregation. The final selected model, ExtraTrees stacking with probability-bias tuning, achieved a test macro-F1 of 0.790, accuracy of approximately 0.921, and weighted-F1 of approximately 0.921.

Academically, the findings show that continuous LLM sentiment scores can function as meta-learning signals because they preserve intensity, disagreement, and boundary information for local calibration.

The study does not claim state-of-the-art performance on UIT-VSFC, since prior transformer- and lexicon-enhanced approaches have reported higher macro-F1. Its contribution is architectural and implementation-oriented: higher education institutions can exploit LLM capabilities through AI-as-a-Service while keeping the locally trained component small, auditable, and adaptable.

The neutral class remained the main unresolved challenge, reinforcing the need for macro-F1, class-level analysis, careful calibration, and future work on uncertainty estimation, aspect-based sentiment analysis, suggestion detection, and human-in-the-loop review for ambiguous cases.

Overall, this study contributes a technically feasible and auditable AIaaS architecture for educational analytics: external LLMs provide semantic scoring capacity, while institutions retain local control over calibration, evaluation, and decision learning. For low-infrastructure higher education settings, the strategic value of AI may lie less in owning large models than in integrating external AI services into locally meaningful and governable analytic systems.

Appendix

A. Reproducibility Materials for the AIaaS Score-stacking Pipeline

1) Scoring prompt structure

Algorithm 2. Required JSON output format for AIaaS scoring

```
{"items":
  [
    {"id":"<original id>",
    "score":<real number from −1 to 1>}
  ]
}
```

Table A1. Calibration anchors used in the AIaaS scoring prompt

ID	Reference sentiment	Score	Anchor sentence
train_000000	Positive	1.0	slide giáo trình đầy đủ. (the lecture slides are complete)
train_000001	Positive	1.0	nhiệt tình giảng dạy, gần gũi với sinh viên. (enthusiastic in teaching and approachable to students)
train_000002	Negative	−1.0	đi học đầy đủ full điểm chuyên cần. (attending classes regularly earns full attendance points)
train_000003	Negative	−1.0	chưa áp dụng công nghệ thông tin và các thiết bị hỗ trợ cho việc giảng dạy. (information technology and supporting equipment have not yet been applied to teaching)
train_000006	Neutral	0.0	em sẽ nợ môn này, nhưng em sẽ học lại ở các học kỳ kế tiếp. (I will fail this course, but I will retake it in subsequent semesters)
train_000012	Neutral	0.0	đang dạy thầy wzjwz208 đi qua nước ngoài giữa chừng, thầy wzjwz209 dạy thay. (the teacher wzjwz208 currently teaching went abroad midway, and teacher wzjwz209 took over)

Note. The anchors were used only to calibrate the continuous scoring scale and were excluded from all scoring, training, development, and test evaluation records.

2) Prompt template used for AIaaS scoring

Algorithm 3. Prompt template for AIaaS continuous sentiment scoring

```
[Calibration instruction] [a1.1] [MĐ1.2]
The following calibration anchors are taken directly from the UIT-VSFC training set and have been excluded from the data to be scored. They are used only to calibrate the continuous sentiment scale and do not restrict the output to the three values −1, 0, and 1.
[Calibration anchors]
id=<anchor_id>; sentence="<anchor_sentence>";
gold_sentiment=<positive/negative/neutral>;
mapped_score=<anchor_score>
[Input batch]
[
  {
    "id":"<record_id>",
    "sentence":"<Vietnamese_feedback_sentence>"
  }
]
[Required output]
Return only the following JSON object format and nothing else:
{
  "items": [
    {
    "id":"<original_id>",
    "score":<real number from −1 to 1>
    }
  ]
```

Table A2. Scorer-level configuration used for AIaaS continuous scoring

Inference Platform	Model Identifier in Script	Output Control	Retry and Validation Setting
Google AI Studio	gemma-4-31b-it	temperature = 0; application/json response; structured response schema requiring items, id, and score	Up to 12 retry attempts with randomized exponential waiting; failed or identifier-mismatched batches were recursively split and reprocessed
Groq	llama-3.1-8b-instant	temperature = 0; JSON-object response format; max_completion_tokens = 4096	Up to 5 retry attempts with exponential waiting; persistent identifier mismatch stopped execution
Groq	qwen/qwen3-32b	temperature = 0; JSON-object response format; max_completion_tokens = 4096	Up to 5 retry attempts with exponential waiting; persistent identifier mismatch stopped execution

Note. Top-p and seed values were not manually specified in the scoring scripts. All validated scores were normalized to the interval [−1, 1], and returned identifiers were checked against input identifiers before scores were appended to scorer-level output files. Because externally served model behavior may change when providers update deployed models, the model identifiers in this table should be interpreted together with the scoring execution date, May 24, 2026, saved scorer-level outputs, and validation records.

Table A3. Module-level input and output of the AIaaS score-stacking pipeline

Module	Main input	Main output
Dataset preparation	UIT-VSFC records with sentence identifiers, texts, gold labels, and official split information	Prepared non-anchor dataset and six removed calibration anchors
Prompt construction	Calibration anchors and batched feedback records	Batch-specific scoring prompt with record identifiers and sentence texts
AIaaS continuous scoring	Scoring prompt and scorer-specific inference configuration	Raw JSON response containing sentence identifiers and continuous sentiment scores
Response validation and normalization	Raw JSON response from each scorer	Validated, identifier-aligned scores clipped to [−1, 1]
Score storage and merging	Validated scorer-level outputs and official split/gold-label file	Merged score table containing Gemma, Llama, and Qwen scores for each non-anchor record
Score-derived feature construction	Merged three-scorer score table	Raw-score, aggregate, disagreement, and interaction features
Local decision calibration and evaluation	Score-derived features, gold labels, and train-development-test split	Selected local decision model, held-out test predictions, metrics, confusion matrices, bootstrap results, and error-analysis files

DATA AND CODE AVAILABILITY

The UIT-VSFC corpus used in this study is publicly available from its original repository. The scoring scripts, calibration anchors, scorer-level outputs, merged predictions, model-selection records, and analysis artifacts are available from the corresponding author upon reasonable request. Exact wall-clock processing time and the invalid-response

rate were not retained as independent runtime artifacts.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Huy-Minh Dau conceptualized the study, designed the methodological framework, developed and implemented the LLM-based sentiment analysis pipeline, processed the dataset, conducted the empirical analysis, interpreted the results, and drafted the manuscript. Thi-Phan-Thu Pham contributed to the literature review, the educational management framing, the interpretation of managerial implications, and the critical revision of the manuscript. All authors discussed the findings, revised the manuscript, and approved the final version for submission.

FUNDING

This research is funded by University of Social Sciences and Humanities, Viet Nam National University Ho Chi Minh City under grant number SĐH2025-07.

REFERENCES

[1] A. Rabelo, M. W. Rodrigues, C. Nobre, S. Isotani, and L. Zárate, "Educational data mining and learning analytics: A review of educational management in e-learning," *IDD*, vol. 52, no. 2, pp. 149–163, 2024. doi: 10.1108/IDD-10-2022-0099

[2] R. Baragash and H. Aldowah, "Sentiment analysis in higher education: A systematic mapping review," *J. Phys.: Conf. Ser.*, vol. 1860, no. 1, 012002, 2021. doi: 10.1088/1742-6596/1860/1/012002

[3] F. Dalipi, K. Zdravkova, and F. Ahlgren, "Sentiment Analysis of Students' Feedback in MOOCs: A Systematic Literature Review," *Front. Artif. Intell.*, vol. 4, 728708, 2021. doi: 10.3389/frai.2021.728708

[4] J. Zhou and J. Ye, "Sentiment analysis in education research: a review of journal publications," *Interactive Learning Environments*, vol. 31, no. 3, pp. 1252–1264, 2023. doi: 10.1080/10494820.2020.1826985

[5] K. V. Nguyen, V. D. Nguyen, P. X. V. Nguyen, T. T. H. Truong, and N. L.-T. Nguyen, "UIT-VSFC: Vietnamese Students' Feedback Corpus for Sentiment Analysis," in *Proc. 2018 10th International Conference on Knowledge and Systems Engineering (KSE)*, 2018, pp. 19–24. doi: 10.1109/KSE.2018.8573337

[6] V. D. Nguyen, K. V. Nguyen, and N. L.-T. Nguyen, "Variants of Long Short-Term Memory for Sentiment Analysis on Vietnamese Students' Feedback Corpus," in *Proc. 2018 10th International Conference on Knowledge and Systems Engineering (KSE)*, 2018, pp. 306–311. doi: 10.1109/KSE.2018.8573351

[7] V. K. C. Bumgardner *et al.*, "Institutional Platform for Secure Self-Service Large Language Model Exploration," *AMIA Jt Summits Transl. Sci. Proc.*, vol. 2025, pp. 105–114, 2025.

[8] P. Mithun, E. Noriega-Atala, N. Merchant, and E. Skidmore, "A Gateway for Egalitarian Access to LLM Based Resources," in *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium, Blue Sky, and WideAIED*, A. I. Cristea, E. Walker, Y. Lu, O. C. Santos, and S. Isotani, Eds. in *Communications in Computer and Information Science*, vol. 2590, Cham: Springer Nature Switzerland, 2025, pp. 137–146. doi: 10.1007/978-3-031-99261-2_13

[9] T. K. F. Chiu, "Future research recommendations for transforming higher education with generative AI," *Computers and Education: Artificial Intelligence*, vol. 6, 100197, 2024. doi: 10.1016/j.caeai.2023.100197

[10] Y. Jin, L. Yan, V. Echeverria, D. Gašević, and R. Martinez-Maldonado, "Generative AI in higher education: A global perspective of institutional adoption policies and guidelines," *Computers and Education: Artificial Intelligence*, vol. 8, 100348, 2025. doi: 10.1016/j.caeai.2024.100348

[11] L. Yan *et al.*, "Practical and ethical challenges of large language models in education: A systematic scoping review," *Brit. J. Educational Tech.*, vol. 55, no. 1, pp. 90–112, 2024. doi: 10.1111/bjet.13370

[12] M. S. Akhtar, A. Ekbal, and E. Cambria, "How Intense Are You? Predicting Intensities of Emotions and Sentiments using Stacked Ensemble [Application Notes]," *IEEE Comput. Intell. Mag.*, vol. 15, no. 1, pp. 64–75, 2020. doi: 10.1109/MCI.2019.2954667

[13] B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*, 1st ed. Cambridge University Press, 2015. doi: 10.1017/CBO9781139084789

[14] C. Zucco, B. Calabrese, and M. Cannataro, "Sentiment analysis and affective computing for depression monitoring," in *Proc. 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Kansas City, MO: IEEE, Nov. 2017, pp. 1988–1995. doi: 10.1109/BIBM.2017.8217966

[15] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992. doi: 10.1016/S0893-6080(05)80023-1

[16] K. M. Ting and I. H. Witten, "Issues in Stacked Generalization," *Journal of Artificial Intelligence Research*, vol. 10, pp. 271–289, 1999. doi: 10.1613/jair.594

[17] A. Bhowmik, N. M. Noor, Md. Mazid-Ul-Haque, M. S. U. Miah, and D. Karmaker, "Evaluating Teachers' Performance through Aspect-Based Sentiment Analysis," in *Proc. 2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, Pune, India: IEEE, 2024, pp. 1–6. doi: 10.1109/I2CT61223.2024.10543706

[18] Y. C. Hua, P. Denny, J. Wicker, and K. Taskova, "A systematic review of aspect-based sentiment analysis: domains, methods, and trends," *Artif. Intell. Rev.*, vol. 57, no. 11, 296, 2024. doi: 10.1007/s10462-024-10906-z

[19] P. X. V. Nguyen, T. T. T. Hong, K. V. Nguyen, and N. L.-T. Nguyen, "Deep Learning versus Traditional Classifiers on Vietnamese Students' Feedback Corpus," arXiv Preprint, arXiv:1911.07223, 2019. doi: 10.48550/arXiv.1911.07223

[20] A. L. Doan and S. T. Luu, "Improving Sentiment Analysis by Emotion Lexicon Approach on Vietnamese Texts," in *Proc. 2022 International Conference on Asian Language Processing (IALP)*, 2022, pp. 39–44. doi: 10.1109/IALP57159.2022.9961318

[21] I. C. Peláez-Sánchez, D. Velarde-Camaqui, and L. D. Glasserman-Morales, "The impact of large language models on higher education: Exploring the connection between AI and Education 4.0," *Front. Educ.*, vol. 9, 1392091, 2024. doi: 10.3389/feduc.2024.1392091

[22] A. Shabir, H. Herwin, A. Asriadi, R. Nurhayati, L. D. Prasojo, and S. C. Dahalan, "Integration of Artificial Intelligence in Virtual Reality-Based Learning," *Data and Metadata*, vol. 4, 859, 2025. doi: 10.56294/dm2025859

[23] M. J. Parker, C. Anderson, C. Stone, and Y. Oh, "A Large Language Model Approach to Educational Survey Feedback Analysis," *International Journal of Artificial Intelligence in Education*, vol. 35, no. 2, pp. 444–481, 2025. doi: 10.1007/s40593-024-00414-0

[24] D. V. Thin, D. N. Hao, and N. L.-T. Nguyen, "Prompt Engineering with Large Language Models for Vietnamese Sentiment Classification," in *Proc. the 38th Pacific Asia Conference on Language, Information and Computation*, N. Oco, S. N. Dita, A. M. Borlongan, and J.-B. Kim, Eds. Tokyo, Japan: Tokyo University of Foreign Studies, 2024, pp. 181–192. https://aclanthology.org/2024.paclic-1.17/

[25] N. X. Phuc, P. N. Xuan, V.-T. Nguyen, T. D. Van, and N. L.-T. Nguyen, "EduPulse: A Practical LLM-Enhanced Opinion Mining System for Vietnamese Student Feedback in Educational Platforms," in *Proc. the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, Y. Matusevych, G. Eryiğit, and N. Aletras, Eds. Rabat, Morocco: Association for Computational Linguistics, 2026, pp. 338–365. doi: 10.18653/v1/2026.eacl-industry.25

[26] J. Geng, F. Cai, Y. Wang, H. Koeppl, P. Nakov, and I. Gurevych, "A Survey of Confidence Estimation and Calibration in Large Language Models," in *Proc. the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Mexico City, Mexico: Association for Computational Linguistics, 2024, pp. 6577–6595. doi: 10.18653/v1/2024.naacl-long.366

[27] J. Cao, S. Kwong, R. Wang, X. Li, K. Li, and X. Kong, "Class-specific soft voting based multiple extreme learning machines ensemble," *Neurocomputing*, vol. 149, pp. 275–284, 2015. doi: 10.1016/j.neucom.2014.02.072

[28] J. Opitz, "A Closer Look at Classification Evaluation Metrics and a Critical Reflection of Common Evaluation Practice," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 820–836, 2024. doi: 10.1162/tacl_a_00675

[29] J. Dong *et al.*, "A Survey on Confidence Calibration of Deep Learning-Based Classification Models Under Class Imbalance Data," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 36, no. 9, pp. 15664–15684, 2025. doi: 10.1109/TNNLS.2025.3565159

[30] Groq on-demand pricing for tokens-as-a-service. *Groq*. [Online]. Available: https://groq.com/pricing
[31] Google AI for Developers. (Jun. 2026). Gemma 4 pricing—Google AI Studio. *Gemini API Documentation*. [Online]. Available: https://ai.google.dev/gemini-api/docs/pricing#gemma-4

Copyright © 2026 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).