

Article

FER-XGBoost: A Hybrid Approach for Fast and Accurate Learner Performance Prediction

Ismail Menyani ^{1,*} , Soukaina Hakkal ¹ , Ahmed Oussous ² and Ayoub Ait Lahcen ^{1,*} 

¹ Engineering Sciences Laboratory, National School of Applied Sciences, Ibn Tofail University, Kenitra 14000, Morocco; soukaina.hakkal@uit.ac.ma

² Department of Informatics, Faculty of Sciences and Techniques of Mohammedia (FSTM), Hassan II University Casablanca, Mohammedia 28800, Morocco; ahmed.oussous@fstm.ac.ma

* Correspondence: ismail.menyani@uit.ac.ma (I.M.); ayoub.aitlahcen@uit.ac.ma (A.A.L.)

Abstract

In adaptive learning environments, accurate prediction of learner performance is essential for delivering personalized educational support. Logistic regression-based models such as PFA, AFM, and DAS3H have demonstrated strong predictive capabilities; however, they face two complementary limitations: high computational cost due to Q-matrix density and a predictive accuracy ceiling inherent to their linear formulation. This study proposes a hybrid algorithm that addresses both limitations simultaneously. The Fast E-learning Recommendation (FER) method is first applied to reduce Q-matrix complexity by aggregating all knowledge components associated with each item into a single combined skill. From an algorithmic perspective, this transformation reduces training time complexity from $O(I \times K)$ to $O(I)$, where I is the number of items and K is the number of knowledge components, yielding substantial computational savings while preserving the conjunctive cognitive structure of the original Q-matrix. XGBoost is then integrated into the simplified models to enhance predictive performance by capturing non-linear relationships that logistic regression cannot capture. The proposed hybrid framework is evaluated on four real-world educational datasets. Experimental results demonstrate that the FER-XGBoost approach achieves a favorable balance between computational efficiency and predictive accuracy: in several configurations, it not only maintains but also slightly improves predictive performance while significantly reducing execution time compared to standard baseline models.

Keywords: performance prediction; adaptive learning; algorithm design; computational complexity; FER; XGBoost; DAS3H; PFA; AFM; intelligent tutorial systems



Academic Editor: Yu-Chen Hu

Received: 20 May 2026

Revised: 22 June 2026

Accepted: 26 June 2026

Published: 26 September 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Computer-Assisted Instruction (CAI), introduced in the 1950s through the “linear programs” of B. F. Skinner [1], represented one of the earliest attempts to integrate technology into education. These early systems provided structured learning sequences but remained limited by their static content, lack of immediate feedback, and inability to adapt to individual learner needs. The emergence of Intelligent Tutoring Systems (ITS) in the 1980s [2] addressed many of these limitations by enabling adaptive instruction based on learner behavior and competency levels [3]. Modern ITS dynamically personalized educational content and recommendations according to learners’ progress, making learner modeling a fundamental component of adaptive learning environments [4–6]. The growing adoption

of adaptive eLearning platforms, as documented in recent evaluations of open-source adaptive learning platforms [7], underscores the increasing need for computationally efficient and pedagogically coherent learner performance models capable of operating in real-time educational environments.

The continuous interaction between learners and ITS generates large volumes of educational data that can be leveraged to predict learner performance and optimize learning trajectories. In this context, knowledge tracing and learner performance prediction have become major research topics in educational data mining. Several statistical and machine learning models have been widely adopted for this purpose, including Performance Factor Analysis (PFA) [8], the Additive Factor Model (AFM) [9,10], the DAS3H model [11], Item Response Theory (IRT) [12], and Bayesian Knowledge Tracing (BKT) [13]. Recent surveys have provided comprehensive overviews of the evolution of these models [14,15]. Most of these approaches rely on a binary representation known as the Q-matrix [16], which maps each item to its associated knowledge components and serves as a fundamental element for assessment and prediction tasks. However, as the number of skills and item-skill associations increase, the Q-matrix becomes increasingly sparse, leading to higher computational complexity and reduced processing efficiency.

Despite the strong predictive capabilities of existing models, two important limitations remain insufficiently addressed. First, the increasing density and structural complexity of the Q-matrix significantly affect training time and computational efficiency, particularly in large-scale and real-time ITS environments. Second, traditional logistic regression-based models such as PFA, AFM, and DAS3H are inherently limited in their ability to capture complex non-linear relationships within learner interaction data. Consequently, most existing studies have focused either on improving predictive accuracy or reducing computational cost, while few approaches effectively address both challenges simultaneously.

To overcome these limitations, this study proposes a hybrid framework that combines structural simplification with ensemble machine learning techniques in order to jointly improve computational efficiency and predictive performance. The proposed approach builds upon two complementary contributions. First, the Fast E-learning Recommendation (FER) method [17] simplifies the Q-matrix by grouping the knowledge components of each item into a single combined skill, reducing matrix density and computational cost. Second, XGBoost [18], a powerful gradient boosting algorithm, is integrated with classical learner modeling approaches such as PFA, AFM, and DAS3H in order to capture non-linear interaction patterns that cannot be effectively modeled using standard logistic regression. In the proposed framework, FER is applied upstream to simplify the Q-matrix structure, while XGBoost is employed downstream to enhance predictive capability.

The proposed hybrid approach is evaluated using several real-world educational datasets collected from different intelligent tutoring systems. Experimental results demonstrate that the proposed framework achieves a favorable trade-off between predictive accuracy and computational efficiency. More specifically, the approach significantly reduces execution time while maintaining, and in some cases improving, predictive performance compared with baseline models. These findings highlight the potential of combining structural optimization techniques with modern machine learning algorithms for scalable and efficient learner modeling in intelligent tutoring systems.

Unlike approaches that introduce new learner modeling paradigms, the objective of this work is to enhance existing and widely adopted logistic regression-based models (PFA, AFM, and DAS3H) through a hybrid framework. The proposed FER-XGBoost framework combines FER-based Q-matrix simplification and XGBoost-based predictive enhancement to jointly address two practical challenges in adaptive learning systems: computational efficiency and predictive performance. Therefore, the contribution of this study lies in the

design and evaluation of a hybrid enhancement framework rather than in the proposal of a fundamentally new learner modeling algorithm.

The main contributions of this work are summarized as follows:

- The design of a hybrid enhancement framework combining FER-based Q-matrix simplification and XGBoost-based predictive enhancement for existing learner performance models (PFA, AFM, and DAS3H);
- A formal complexity analysis of the FER transformation and its impact on reducing the computational cost of Q-matrix-based models;
- An empirical evaluation on four benchmark educational datasets demonstrating the effectiveness of the proposed framework in balancing execution speed and predictive accuracy.

The remainder of this paper is organized as follows. Section 2 reviews related work on learner performance prediction and knowledge tracing models. Section 3 presents the proposed hybrid framework and the FER-based simplification strategy. Section 4 presents and discusses the obtained results, and Section 5 concludes the paper and outlines future research directions.

2. Related Work

2.1. Interpretable Performance Prediction Models in E-Learning: PFA, AFM, and DAS3H

Learner performance modeling plays a crucial role in intelligent educational systems, enabling platforms to monitor student progress, identify knowledge gaps, and adapt instructional strategies [19]. In educational data mining, interpretable models such as logistic regression and rule-based approaches are particularly valued for their transparency, allowing educators to better understand and trust model predictions [20,21]. Recent studies have also investigated neural and attention-based knowledge tracing architectures to enhance long-term modeling of learner states [22,23].

Performance Factor Analysis (PFA) [8] is a logistic regression-based model that explicitly accounts for a learner's history of successes and failures per skill. For each knowledge component k , three parameters are defined: β_k (the difficulty of skill k), and γ_k^f (the learning gain associated with successful attempts on skill k). Given a target item j associated with the set of knowledge components $K(j)$, the probability that student i answers item j correctly is formulated as:

$$p_{ij} = \sigma\left(\sum_{k \in K(j)} \left(\beta_k + \gamma_k^s T_{ik}^s + \gamma_k^f T_{ik}^f\right)\right) \quad (1)$$

where T_{ik}^s and T_{ik}^f denote, respectively, the cumulative number of successful and failed attempts by student i on skill k prior to the current attempt, and σ is the logistic function. PFA has demonstrated competitive predictive performance compared to more complex models, particularly when appropriate tuning strategies are applied [24,25].

The Additive Factor Model (AFM) [9,10] simplifies this framework by assuming a single learning rate per skill, without distinguishing between successes and failures. For each knowledge component k , two parameters are defined: β_k (the difficulty of skill k) and γ_k (the learning rate associated with practice on skill k). The probability that student i answers item j correctly is modeled as:

$$p_{sij} = \sigma(\alpha_s + \sum_{k \in K(j)} \beta_k + \gamma_k N_{ik}) \quad (2)$$

where N_{ik} denotes the cumulative number of practice opportunities (i.e., prior attempts, regardless of outcome) that student i has had on skill k , α_s represents the baseline proficiency (or initial mastery level) of learner s , $K(j)$ is the set of knowledge components associated

with item j , and $\sigma(\cdot)$ is the logistic function defined above. Despite its simplicity, AFM has shown robustness and generalizability across diverse learner datasets [26].

DAS3H [11] extends these frameworks by incorporating temporal dynamics through a multiscale decay scheme that models both learning and forgetting effects:

$$p_{sjt} = \sigma \left(\alpha_s - \delta_j + \sum_k \left(\gamma_k^s \cdot \log T_{skt}^s + \gamma_k^f \cdot \log T_{skt}^f \right) \right) \quad (3)$$

where α_s is student ability, δ_j is item difficulty of item j , and γ_k^s and γ_k^f are the learning-gain parameters associated with successful and failed attempts on skill k , respectively. The terms T_{skt}^s , T_{skt}^f represent the counts of successes and failures of student j on skill k , aggregated over a set of predefined time windows (multiscale decay) up to time t ; the logarithmic transformation $\log(1 + \cdot)$ attenuates the influence of older interactions, thereby implementing the forgetting component of the model. $\sigma(\cdot)$ is the logistic function defined above. DAS3H has shown superior AUC performance on large-scale benchmarks, confirming its scalability and effectiveness in time-aware learning scenarios [11].

Despite their individual strengths, PFA, AFM, and DAS3H share two fundamental limitations that prior work has not jointly addressed. First, their logistic regression formulation imposes a linear predictive ceiling: all three models compute the probability of a correct response as a sigmoid of a weighted sum, which prevents them from capturing higher-order interactions between learner history, item difficulty, and skill mastery [18]. Second, these models rely directly on the raw Q-matrix structure, meaning that as the number of knowledge components associated with items increases, the computational cost of feature construction scales as $O(I \times K)$, rendering them increasingly impractical in large-scale or real-time ITS environments.

Hybrid approaches that integrate ensemble methods such as XGBoost with classical models [18,27] have addressed the first limitation, but none has simultaneously tackled the structural complexity of the Q-matrix. The present work fills this gap by combining FER-based Q-matrix simplification with XGBoost integration in a single two-stage framework. Table 1 provides a comparative summary of their main characteristics.

Table 1. Comparison of PFA, AFM, and DAS3H Models.

Aspect	PFA	AFM	DAS3H
Name	Performance Factors analysis	Additive Factor Model	Item Difficulty, Student Ability, Skill, and Student Skill Practice History
Type	Logistic Regression-based	Logistic Regression-based	Logistic Regression-based with temporal features
Main Components	Skill difficulty, prior successes & failures	Skill difficulty, total practice count	Student ability, item difficulty, skill practice history over time
Accounts for Successes/Failures	Yes	No (aggregated practice only)	Yes
Handles Time/Decay	No	No	Yes (multi-scale temporal decay)

Table 1. Cont.

Aspect	PFA	AFM	DAS3H
Student Ability Modeling	No	No	Yes
Item Difficulty Modeling	No	Optional	Yes
Interpretability	High	High	High
Scalability	High	High	High (with added complexity)
Use Cases	Early modeling of skill-level learning	Simple practice-based prediction	Scenarios requiring time-aware modeling and long-term learning dynamics

2.2. Gradient Boosting Techniques for Model Enhancement in Learning Analytics

Over the past few years, boosting techniques have gained significant attention in the field of educational data mining due to their strong predictive capabilities. Unlike traditional approaches that rely on manually engineered features, these methods iteratively improve performance by focusing on difficult-to-predict instances. Their ensemble-based structure, robustness to overfitting, and ability to model complex data (noisy, sparse, or high-dimensional) make them particularly suitable for analyzing learner interaction data. Recent studies have demonstrated that integrating ensemble learning techniques such as Random Forest, AdaBoost, and XGBoost with classical Knowledge Tracing models, including PFA, can substantially improve learner performance prediction accuracy across multiple educational datasets [27]. This effectiveness of hybrid machine learning architectures in ITS settings has been further confirmed by recent work demonstrating that combining unsupervised techniques such as PCA-based dimensionality reduction and K-Means clustering with supervised classifiers significantly improves student performance prediction across multiple educational datasets [28].

Gradient boosting decision tree (GBDT) algorithms have become one of the most effective ensemble learning approaches for classification and regression tasks. Among the most widely adopted implementations are XGBoost [29] and LightGBM [30], both of which construct an ensemble of decision trees sequentially to minimize a regularized objective function. The general objective function of gradient boosting is defined as:

The Extreme Gradient Boosting (XGBoost) algorithm [29] stands out as an efficient implementation of gradient boosting, sequentially constructing trees to minimize a regularized loss function. The objective function is described in the following way:

$$\mathcal{L}(\Phi) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (4)$$

where:

- $l(y_i, \hat{y}_i)$ is a differentiable loss function;
- f_k represents each regression tree in the ensemble;
- $\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \|w\|^2$ is a regularization term controlling model complexity, with T being the number of leaves and w the leaf weights.

Both XGBoost and LightGBM optimize this objective using second-order gradient information. However, they differ in their tree construction strategies. XGBoost grows trees level by level, producing balanced trees that often improve generalization, whereas LightGBM adopts a leaf-wise growth strategy with depth constraints and histogram-based split finding, resulting in faster training and lower memory consumption, particularly on large-scale and sparse datasets. These complementary characteristics make both algorithms well suited for learner performance prediction tasks involving high-dimensional educational data.

In our previous work [18], we investigated the incorporation of XGBoost to enhance classical logistic regression-based models such as PFA, AFM, and DAS3H. Rather than replacing these interpretable models, XGBoost was employed as a complementary learning layer to capture complex non-linear interactions that cannot be effectively modeled by logistic regression alone. Experimental results demonstrated consistent improvements in predictive performance across multiple educational datasets while preserving the interpretability of the underlying learner models (Figure 1).

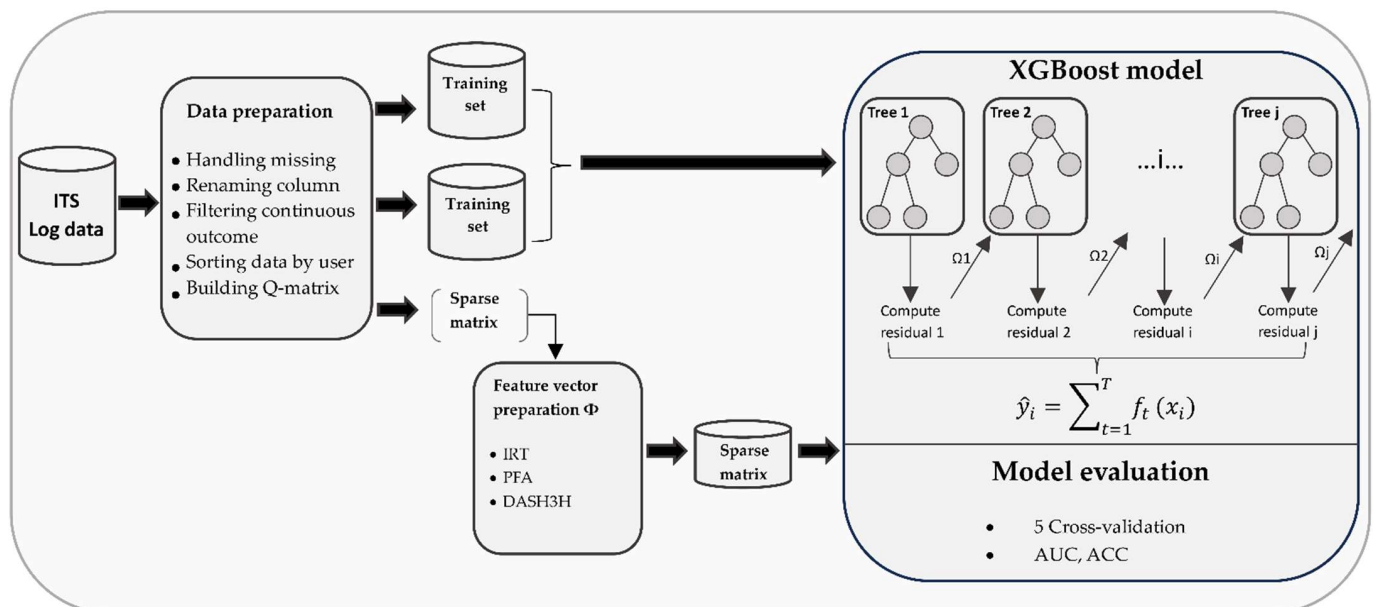


Figure 1. XGBoost learner performance prediction framework.

In the present study, we extend this investigation by additionally evaluating LightGBM as an alternative gradient boosting framework. This comparison aims to determine whether the predictive improvements observed with XGBoost are specific to its boosting strategy or can be generalized to another state-of-the-art gradient boosting algorithm. Furthermore, comparing XGBoost and LightGBM enables us to analyze the trade-off between predictive performance and computational efficiency across different boosting implementations.

Table 2 Summary of Gradient Boosting Integration to the improvement of classical models in our framework, highlighting feature design, calibration strategy, and observed improvements.

Table 2. Summary of Gradient Boosting Integration for Model Enhancement in Learning Analytics.

Aspect	XGBoost	LightGBM
Base Models Enhanced	PFA, AFM, DAS3H	PFA, AFM, DAS3H
Tree growth	Level-wise	Leaf-wise
Split strategy	Exact/Approximate	Histogram-based
Sparse data	Good	Excellent
Training speed	Moderate	Faster
Main advantage	High predictive accuracy	High accuracy with lower computational cost
Type of Features Used	learner interaction data: history of successes and failures, time, and skill	
Main Contribution	Integration of XGBoost/LightGBM to enhance predictive performance while preserving the structure of interpretable models	
Justification for Use	addresses the inability of baseline models to fully capture non-linear relationships and interactions without affecting interpretability.	

2.3. Deep and Attention-Based Knowledge Tracing Models: From DKT to Transformer Architectures

The limitations of logistic regression-based models motivated the development of deep learning approaches to knowledge tracing, beginning with Deep Knowledge Tracing (DKT) [31], which applies Long Short-Term Memory (LSTM) networks to model the temporal evolution of learner knowledge states. DKT demonstrated significant AUC improvements over BKT and PFA on standard benchmarks, but at the cost of interpretability: the latent knowledge state is not directly mappable to individual skills or pedagogical constructs.

Dynamic Key-Value Memory Networks (DKVMN) [32] extended this line of work by introducing an explicit memory structure that stores and updates skill-level knowledge representations. While more interpretable than DKT, DKVMN still requires a predefined number of latent concepts and does not naturally accommodate the Q-matrix encoding used in PFA, AFM, or DAS3H.

Self-Attentive Knowledge Tracing (SAKT) [33] introduced the first self-attention mechanism for knowledge tracing, computing relevance weights between past interactions and the current question. Subsequent models Attentive Knowledge Tracing (AKT) [34] and SAINT [35], further refined this approach by incorporating monotonic attention and encoder-decoder architectures respectively, achieving state-of-the-art performance on several benchmarks. Graph-based models such as GKT [36] explicitly model the dependency structure between knowledge concepts, offering a complementary perspective.

Despite these advances, deep and transformer-based KT models present several limitations in the context of our problem setting. First, they require large volumes of interaction data to train reliably, a constraint that may not be satisfied in small or sparse educational datasets. Second, their black-box nature conflicts with the interpretability requirements of adaptive learning systems, where educators need to understand and trust model decisions. Third, and most critically for our work, these architectures are not designed to exploit the Q-matrix structure directly: they either bypass it entirely (DKT, DKVMN) or treat it as a simple input embedding (SAKT, AKT), and none of them addresses the $O(I \times K)$ computational bottleneck that Q-matrix density introduces in classical models. The hybrid FER-XGBoost approach proposed in this study is therefore positioned as a complement to this family of models: it preserves the pedagogical transparency and Q-matrix grounding

of PFA/AFM/DAS3H while closing the gap in predictive performance through gradient boosting, without the data hunger and opacity of deep architectures.

2.4. Complexity Reduction in Knowledge Tracing Models Using Q-Matrix Simplification (FER)

A major challenge in learning analytics is the computational cost induced by the Q-matrix [16], which encodes item–skill relationships and becomes denser as the number of associated skills increases, leading to higher computational costs, particularly in large-scale or real-time environments [37].

To address this, the Fast E-learning Recommendation (FER) approach simplifies the Q-matrix by grouping all skills associated with a given item into a single combined skill, thereby reducing item–skill associations and accelerating model training and inference [17].

Unlike traditional dimensionality reduction techniques such as feature hashing or matrix factorization [37], FER operates directly on the knowledge representation level and preserves the conjunctive cognitive structure of items: a learner who masters the combined skill is assumed to have practiced the full original set of knowledge components. However, this transformation does not preserve individual skill-level diagnostic resolution the fine-grained per-skill mastery history is replaced by a single aggregated signal. This is a known and bounded limitation of the approach [17]: in datasets where individual skill trajectories carry high discriminative value (e.g., datasets without temporal information, or with high skill diversity), a modest AUC degradation may occur. This trade-off is partially recoverable through the XGBoost stage of the pipeline, as demonstrated in the results. By reducing the structural burden of the Q-matrix, FER enables classical models to scale more efficiently while maintaining comparable predictive performance. Experimental results have demonstrated that this approach significantly improves execution time without substantial loss in accuracy, making it particularly suitable for adaptive learning systems requiring both responsiveness and reliability [17].

The FER transformation operates in two stages. In the first stage, the original Q-matrix $Q \in \{0, 1\}^{I \times K}$ encodes the relationship between items and knowledge components: an entry $q_{ik} = 1$ indicates that item i requires skill k , and its complexity is measured by the total number of such associations, $\text{Complexity}(Q) = \sum_i \sum_k q_{ik}$. As the number of multi-skill items grows, this density directly translates into higher computational costs during model training. In the second stage, FER reduces this complexity by grouping all skills associated with a given item into a single combined skill $c_j = k_1 \wedge k_2 \wedge \dots \wedge k_{n_j}$, yielding a transformed matrix Q' where each item maps to exactly one component. As a result, the average number of knowledge components per item drops to 1, regardless of the original skill density.

Algorithmic complexity analysis. Let I denote the number of items and K the number of knowledge components. The original Q-matrix $Q \in \{0, 1\}^{I \times K}$ incurs a training complexity of $O(I \times K)$ for feature construction, since each item is associated with up to K skills. After FER transformation, the resulting matrix $Q' \in \{0, 1\}^{I \times 1}$ reduces this complexity to $O(I)$, as each item maps to exactly one combined skill. Consequently, the FER algorithm achieves a complexity reduction factor of K which, for datasets such as Algebra05 ($K = 112$) and Bridge-Algebra06 ($K = 493$), represents a significant theoretical gain, consistent with the empirical time reductions observed experimentally.

This transformation does not discard pedagogical information: the combined skill c_j encapsulates the conjunction of the original skills, so a learner who masters c_j is assumed to have acquired the full set $\{k_1, k_2, \dots, k_{n_j}\}$. Crucially, because many items in real-world datasets share the same skill combinations, the redundancy present in the data mitigates the loss of individual skill granularity. The complete pipeline of this approach from raw Q-matrix to the simplified sparse learning matrix fed into the FER-DAS3H, FER-PFA, and

FER-AFM models is illustrated in Figure 2, which shows how knowledge component combination and data encoding work together to reduce computational overhead while preserving the adaptive learning capability of the system.

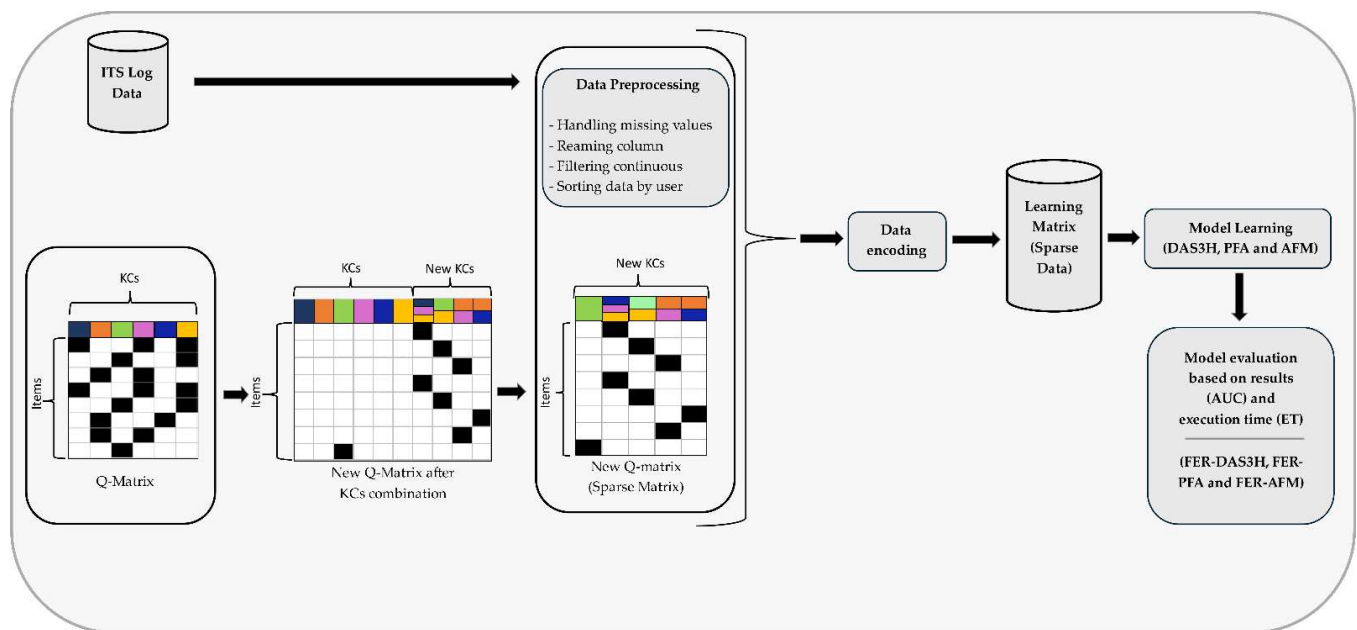


Figure 2. FER-MODEL Learner performance prediction framework.

2.5. Hybrid Approach Combining XGBoost and Q-Matrix Optimization Techniques for Adaptive Learning

Building on the strengths and limitations of interpretable models such as PFA, AFM, and DAS3H, as well as the performance gains achieved by integrating XGBoost, we propose a hybrid approach that simultaneously addresses predictive accuracy and computational efficiency in adaptive learning environments. This approach combines two complementary strategies: XGBoost enhances logistic regression-based models by capturing non-linear interactions among features, improving predictive performance beyond what interpretable models can achieve alone, while the Q-matrix simplification introduced in Section 2.4 reduces computational cost without significantly sacrificing interpretability. Together, these strategies offer a balanced solution: XGBoost contributes predictive power, while Q-matrix simplification ensures computational efficiency.

The hybrid framework was empirically evaluated on multiple real-world educational datasets. The results show that this combination not only improves predictive performance metrics (AUC, accuracy) compared to classical models, but also significantly reduces execution time. Importantly, the gains in speed are achieved without compromising interpretability, making the approach suitable for practical adaptive learning systems where both reliability and responsiveness are crucial.

Integration with XGBoost (Global Equation)

The integration logic follows a sequential two-stage architecture. In the first stage, the FER transformation is applied to the Q-matrix to produce the simplified feature representation $\phi_{FER}(x)$. This compressed feature space preserves the essential item–skill associations while substantially reducing matrix density, thereby lowering the computational burden of the downstream models PFA, AFM, and DAS3H. The logistic base $P(y = 1 | x) = \sigma(WL\phi_{FER}(x))$ then provides an interpretable, linear estimate of the proba-

bility that a learner answers an item correctly, grounded in the pedagogical structure of the Q-matrix.

In the second stage, XGBoost is applied to the same FER-transformed features as a complementary predictive layer. XGBoost constructs an ensemble of T -boosted decision trees, where each tree $f_t \in \mathcal{F}$ is fitted sequentially on the residuals of the previous model, progressively correcting prediction errors. The aggregate score $\hat{z}_i = \sum_{t=1}^T f_t(x_i)$ captures non-linear interactions between student history, item difficulty, and skill mastery that lie beyond the reach of logistic regression. The final hybrid prediction $P(y = 1 | x) = \sigma(WL\phi_{FER}(x) + \sum_{t=1}^T f_t(x_i))$ thus combines interpretability with predictive power: the linear term provides a pedagogically grounded baseline, while the XGBoost term corrects systematic biases in a data-driven fashion. The full data flow of this pipeline includes ITS log data through data preparation, sparse matrix construction, feature encoding, and model evaluation.

It should be noted that the experimental evaluation presented in Section 4 is intentionally scoped to the logistic regression-based model family (PFA, AFM, DAS3H), as FER-XGBoost is specifically designed to enhance this class of models. A comparison with deep knowledge tracing models such as DKT, DKVMN, SAKT, AKT, and SAINT constitutes a natural and planned extension of this work, as discussed in Section 5.

3. Methodology

3.1. Datasets

To assess the effectiveness and generalizability of the proposed hybrid approach, we performed experiments on four widely used real-world educational datasets, each providing diverse learning contexts and interaction patterns: Algebra05, Bridge to Algebra06, Assistments 2009, and Assistments 2017 (Table 3).

Table 3. Datasets characteristics.

Dataset	Learners	Items	Skills(KCs)	Interactions	Timestamps	KCs per Item
Algebra05	574	173,113	112	606,983	Yes	1.363
Bridge-Algebra06	1146	129,263	493	1,817,393	Yes	1.014
Assist09	3241	17,709	124	278,336	No	1.197
Assist17	1708	3162	102	934,638	Yes	1.225

Algebra05: The Algebra 2005 dataset is part of the KDD Cup 2010 collection [38]. It involves interactions between students and an intelligent tutoring system for high school algebra. It contains logs of problem-solving steps, timestamps, problem identifiers, and correction labels. It is extensively used to evaluate models that predict student performance across time.

Bridge to Algebra06: The Bridge to Algebra 2006 dataset is another KDD Cup 2010 dataset, targeting secondary school students working with a program designed to prepare them for algebra [38]. It includes student responses, skill labels, and temporal information, providing a valuable input for sequential modeling and performance prediction assignments.

Assistments 2009: The ASSISTments 2009 dataset is a log of student interactions with the ASSISTments online tutoring platform [39]. It includes anonymous student IDs, item IDs, skill labels, timestamps, and response accuracy. It allows researchers to build models of skill mastery and learning progression.

Assistments 2017: The ASSISTments 2017 dataset is a newer, structured version of the ASSISTments logs [40]. It is cleaned and curated for research use, including fields such as user ID, skill ID, problem ID, timestamp and binary correction. It is often used to assess deep learning and ensemble models in educational contexts, due to its size and coherence.

3.2. Experiment

3.2.1. Preprocessing

Data preprocessing is a crucial step to ensure the quality and relevance of the obtained results. The preprocessing pipeline was applied consistently across all four datasets and comprised four sequential stages, as summarized in Table 4.

Stage 1 Removal of records with missing skill tags. Interactions without an associated knowledge component label were systematically discarded, as skill information is required by all three base models (PFA, AFM, DAS3H).

Stage 2 Exclusion of learners with insufficient interaction history. Learners with fewer than 10 interactions in total were excluded from the analysis, as such short sequences do not provide sufficient evidence for reliable skill mastery estimation. This threshold is consistent with standard practice in the knowledge tracing literature [11,40].

Stage 3 Filtering of inconsistent or duplicate records. Records were flagged as inconsistent when they contained response values outside the binary set {0, 1}, or when the same learner–item–timestamp triplet appeared more than once. All such records were removed prior to model training.

Stage 4 Chronological sorting. Remaining interactions were sorted by learner identifier and timestamp to preserve the sequential structure of learning histories, which is required by the temporal features of DAS3H (sliding time-window counters).

Table 4. Summary of preprocessing attrition across the four datasets.

Dataset	Initial Size	Learners < 10 Interactions Excluded	Duplicates Removed	NaN Skill_Id Removed	Total Removed	Final Size	% Removed
Algebra05	809,694	0	0	202,711	202,711	606,983	25.04%
Bridge-Algebra06	3,679,199	0	0	1,861,806	1,861,806	1,817,393	50.60%
Assist09	528,428	4976	17,879	66,326	250,092	278,336	47.33%
Assist17	942,816	1	8176	1	8178	934,638	0.87%

Once this stage is completed, the construction of the Q-matrix is carried out. This matrix formalizes the relationship between items (questions) and the associated skills in the form of a binary structure. To reduce computational complexity, a simplification step is applied, grouping certain skills associated with the same item into a combined skill. This transformation decreases the density of the matrix while preserving the pedagogical coherence and interpretability of the item–skill relationships.

Finally, feature encoding was performed using one-hot encoding to represent learner, item, and skill identifiers. Derived variables such as the cumulative number of successes or failures per skill and sliding time-window counters were computed after sorting and integrated depending on the model: PFA and DAS3H use per-skill success/failure counts, AFM uses the total practice count per skill, and DAS3H additionally requires time-stamped window-based counters.

3.2.2. Proposed Hybrid Approach

The proposed approach is based on a hybrid architecture combining two complementary contributions: on the one hand, a simplification of the Q-matrix to reduce computational complexity, and on the other hand, the integration of the gradient boosting model XGBoost to enhance predictive capacity. This design simultaneously increases efficiency by reducing the structural burden of the item–skill representation and improves prediction accuracy through XGBoost’s ability to capture complex, non-linear relationships in learner data.

The first component of the approach consists of transforming the Q-matrix by associating each item with a single aggregated skill. This operation significantly reduces the density of the matrix and, consequently, the computational cost of classical models such as PFA, AFM, and DAS3H, while preserving the cognitive structure essential for pedagogical interpretation.

The second component involves integrating XGBoost as a complementary predictive engine. Unlike traditional linear logistic models, XGBoost relies on boosted decision trees capable of capturing non-linear interactions between variables. The use of this model therefore enhances prediction accuracy by exploiting the complexity of the learning data. In practice, XGBoost is applied alongside PFA, AFM, and DAS3H models in order to benefit both from the efficiency gained through Q-matrix simplification and from the predictive robustness of an ensemble algorithm.

This hybridization thus meets a dual requirement: reducing model execution time while maximizing predictive performance. In other words, Q-matrix simplification makes the models applicable in near real-time contexts, while XGBoost acts as a performance amplifier on already preprocessed and optimized data. The complete hybrid algorithm is formalized in Algorithm 1.

Algorithm 1: FER-XGBoost Hybrid Training Procedure

Input: Raw interaction logs L , Q-matrix Q , number of boosting iterations T

Output: Trained hybrid model H

1. Preprocess L : remove missing values, sort by learner and timestamp
 2. **FER Transformation:** For each item $i \in I$, group all associated skills $\{k_1, k_2, \dots, k_n\}$ into combined skill $c_i = k_1 \wedge k_2 \wedge \dots \wedge k_n \rightarrow$ produce Q'
 3. Encode features $\phi_{FER}(x)$ using Q' (one-hot encoding of learner, item, combined skill)
 4. **Base model training:** Fit logistic model $P(y = 1 \mid x) = \sigma(WL\phi_{FER}(x))$ on training set
 5. **XGBoost training:** For $t = 1$ to T do: fit tree f_t on residuals of previous ensemble
 6. **Combine:** $\hat{y} = \sigma(WL\phi_{FER}(x) + \sum_{t=1}^T f_t(x_i))$
 7. **Return** $H = \{WL, f_1, \dots, f_T\}$
-

Figure 3 below provides a comprehensive visual representation of the complete FER-XGBoost pipeline, illustrating the sequential data flow from raw interaction logs to the final hybrid prediction model. The framework unfolds across six interdependent stages.

In the first stage, raw interaction data collected from ITS logs comprising learner identifiers, item identifiers, skill labels, binary responses, and timestamps constitute the input to the system.

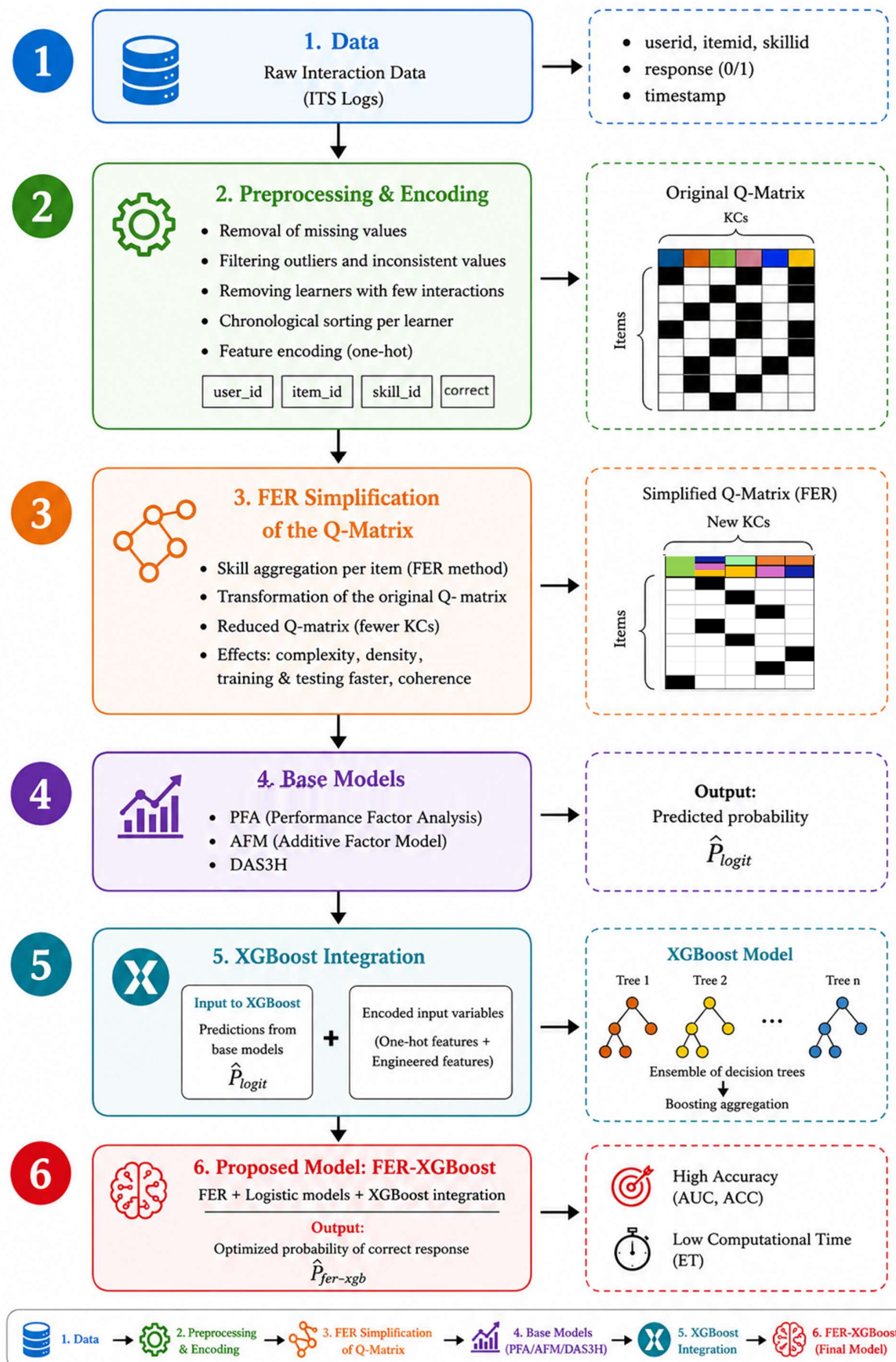


Figure 3. FER-XGBoost MODEL Learner performance prediction framework.

The second stage encompasses data preprocessing and feature encoding: missing values and inconsistent records are removed, interactions are sorted chronologically by learner, and the resulting cleaned data are encoded via one-hot representations of learner, item, and skill identifiers, supplemented by derived variables such as cumulative success and failure counts per skill.

In the third stage, the FER transformation is applied to the original Q-matrix, aggregating all knowledge components associated with each item into a single combined skill, thereby producing a structurally simplified matrix Q' that substantially reduces computational overhead while preserving pedagogical coherence.

This simplified representation is then fed into the fourth stage, where classical logistic regression-based models PFA, AFM, and DAS3H are trained to produce an interpretable linear estimate of the probability of a correct learner response, denoted $\hat{p}^{\text{logit}}$.

In the fifth stage, XGBoost is applied as a complementary predictive layer, receiving both the logistic model predictions and the FER-encoded feature space as input, and constructing a sequential ensemble of boosted decision trees to capture non-linear interaction patterns that lie beyond the reach of logistic formulations.

Finally, the sixth stage yields the FER-XGBoost model, whose output $\hat{P}_{fer+ xgb}$ represents the optimized probability of a correct response, combining the interpretability of the logistic baseline with the predictive power of gradient boosting. This end-to-end architecture ensures that computational efficiency and predictive accuracy are jointly addressed at each stage of the pipeline.

3.2.3. Evaluation Methodology

The proposed models were evaluated using a 5-fold cross-validation at the student level, following the protocol established in prior work on learner performance prediction [11,17,18]. The learner population in each dataset was partitioned into 5 disjoint subsets based on learner identifiers. In each fold, 4 subsets were used for training and the remaining subset for testing. Crucially, this student-level partitioning ensures that all interactions belonging to a given learner are assigned exclusively to either the training set or the test set, with no overlap between the two. This design explicitly prevents data leakage: unlike random interaction-level splitting which can allow future interactions of a learner to appear in the training set while earlier ones are used for evaluation student-level cross-validation respects the sequential and personalized nature of learner interaction data. The reported AUC, ACC, and execution time (TE) values represent averages computed across the 5 folds, and this protocol was applied consistently to all model configurations and datasets.

The XGBoost model was implemented using the XGBoost library in Python with the following hyperparameter configuration, consistent with the validated setup reported in our prior work [18]. The ensemble comprises 1000 boosted decision trees, which provides sufficient capacity to reduce training bias without inducing overfitting. A learning rate of 0.1 (shrinkage) was applied to improve generalization by scaling the contribution of each tree, at the cost of moderately increased training time. An L2 regularization parameter of $\text{reg-lambda} = 1.0$ was used to penalize large leaf weights, controlling model complexity and reducing variance on unseen learner data. No early stopping criterion was applied, as this combination of iteration count and regularization was found to yield stable convergence across all datasets. All remaining hyperparameters were set to their XGBoost library default values: maximum tree depth ($\text{max_depth} = 6$), subsample ratio of training instances ($\text{subsample} = 1.0$), subsample ratio of columns per tree ($\text{colsample_bytree} = 1.0$), minimum sum of instance weight required in a child node ($\text{min_child_weight} = 1$), and minimum loss reduction required for a further partition ($\text{gamma} = 0$). A fixed random seed of 42

was used throughout the training process to ensure the reproducibility of the experimental results. This initialization guarantees consistent model training across repeated executions while maintaining the stability of the reported metrics throughout the 5-fold student-level cross-validation described above. The optimal configuration was identified through the 5-fold student-level cross-validation protocol described above, and all reported metrics correspond to this best-selected configuration.

All experiments were conducted on a Windows 11 machine equipped with 16 GB of RAM and an Intel Core i7 11th-generation processor, without GPU acceleration. Models were implemented in Python 3.11.4 using scikit-learn for logistic regression and the XGBoost Python library for the gradient boosting stage. Reported execution times were obtained under single-threaded execution conditions.

To evaluate both the predictive capability and the efficiency of the proposed models, three evaluation measures were considered: the area under the ROC curve (AUC), accuracy (ACC), and execution time in seconds (TE).

- AUC measures the model's ability to discriminate between correct and incorrect responses and provides a threshold-independent assessment of classification performance. A higher AUC value indicates better discrimination ability.
- Accuracy (ACC), expressed as the proportion of correct predictions relative to the total number of predictions, provides an intuitive measure of the overall correctness of the model.
- Root Mean Squared Error (RMSE) quantifies the average magnitude of the difference between predicted probabilities and actual binary outcomes. Unlike AUC and ACC, which assess discrimination or classification correctness, RMSE provides a measure of the calibration and precision of the predicted probabilities themselves. A lower RMSE value indicates predictions that are closer to the true outcomes.
- Negative Log-Likelihood (NLL) evaluates the quality of the predicted probability distribution by penalizing confident but incorrect predictions more heavily than uncertain ones. As a proper scoring rule, NLL provides a complementary, probability-sensitive measure of model performance. A lower NLL value indicates better-calibrated and more reliable probabilistic predictions.
- Execution time (TE), expressed in seconds, refers specifically to the duration of the model training stage (i.e., the fit operation), measured from the start to the end of the model's parameter estimation on the training set for a given fold. It does not include data loading, preprocessing, feature generation (including FER-based Q-matrix simplification), or inference on the test set, which are common across all configurations and datasets and are therefore not included in the comparative timing analysis. This metric is particularly important in real-time applications, where system responsiveness is as critical as predictive accuracy.

These measures were computed on a test set selected for each dataset to ensure a fair comparison between the baseline models and the proposed approach. The evaluation was carried out on several real-world datasets from various intelligent tutoring systems. This diversity ensures the generalization of the results and makes it possible to verify the consistency of the approach in terms of both predictive performance and computational efficiency across different pedagogical and disciplinary contexts.

It should be noted that the experimental comparison presented in this study is intentionally scoped to the logistic regression-based model family (PFA, AFM, DAS3H) and their FER and XGBoost variants. This design choice reflects the core contribution of the work: demonstrating that Q-matrix structural simplification (FER) combined with gradient boosting (XGBoost) yields a favorable trade-off between computational efficiency and predictive accuracy within this specific model family. Comparison with tree-based ensemble

alternatives such as Random Forest, LightGBM, or CatBoost would constitute a separate experimental contribution beyond the scope of the present study. Comparison with deep and attention-based knowledge tracing models such as DKT or AKT is architecturally incompatible with the FER transformation, which operates specifically on the discrete item–skill structure of the Q-matrix a structure that neural KT models either bypass entirely or flatten into dense embeddings. Furthermore, prior empirical work has demonstrated that logistic regression with well-engineered features achieves performance comparable to DKT on standard benchmarks [40], confirming the competitiveness of the baseline models used in this study.

4. Results & Discussion

The following Tables 5–8 provide a comparative assessment of the basic models and their improved versions using the proposed hybrid approach, evaluated on all datasets according to the AUC and Accuracy measures.

All evaluation tables (Tables 5–8) report mean $\pm$ standard deviation across the 5 student-level cross-validation folds, providing a direct indication of the stability of each model configuration. The standard deviations remain consistently small across all datasets and model families (typically below 0.01 AUC, e.g., DAS3H-LR on Bridge-Algebra06: 0.790 ± 0.0015 ; PFA-XGBoost-FER on Assist17: 0.662 ± 0.0037), indicating that the reported performance estimates are stable and not driven by fold-specific variability.

Table 5. Algebra05 evaluation results (mean $\pm$ standard deviation across 5 cross-validation folds).

Algebra05	ACC	AUC	RMSE	NLL	TE
DAS3H-LR	0.817 ± 0.0056	0.826 ± 0.0029	0.3626 ± 0.0047	0.4128 ± 0.0087	36.00
DAS3H-LR-FER	0.816 ± 0.0059	0.829 ± 0.0028	0.3606 ± 0.0044	0.4093 ± 0.0081	12.07
DAS3H-XGBoost	0.812 ± 0.0072	0.822 ± 0.0041	0.3668 ± 0.0056	0.4219 ± 0.0103	127.2
DAS3H-XGBoost-FER	0.811 ± 0.0082	0.821 ± 0.0035	0.3693 ± 0.0059	0.4266 ± 0.0109	90.2
DAS3H-LightGBM	0.813 ± 0.0065	0.819 ± 0.0040	0.3652 ± 0.0049	0.4189 ± 0.0088	25.0
DAS3H-LightGBM-FER	0.810 ± 0.0070	0.815 ± 0.0037	0.3680 ± 0.0053	0.4239 ± 0.0095	16.0
PFA-LR	0.788 ± 0.0079	0.744 ± 0.0076	0.3910 ± 0.0045	0.4711 ± 0.0082	15.95
PFA-LR-FER	0.789 ± 0.0076	0.748 ± 0.0069	0.3905 ± 0.0045	0.4697 ± 0.0083	4.19
PFA-XGBoost	0.791 ± 0.0067	0.770 ± 0.0066	0.3856 ± 0.0046	0.4599 ± 0.0090	22.03
PFA-XGBoost-FER	0.785 ± 0.0068	0.770 ± 0.0060	0.3879 ± 0.0045	0.4641 ± 0.0089	12.35
PFA-LightGBM	0.796 ± 0.0057	0.775 ± 0.0072	0.3822 ± 0.0039	0.4523 ± 0.0073	6.9
PFA-LightGBM-FER	0.791 ± 0.0060	0.771 ± 0.0064	0.3861 ± 0.0040	0.4601 ± 0.0075	7.1
AFM-LR	0.781 ± 0.0086	0.723 ± 0.0041	0.3979 ± 0.0064	0.4852 ± 0.0123	5.94
AFM-LR-FER	0.785 ± 0.0089	0.727 ± 0.0037	0.3975 ± 0.0064	0.4848 ± 0.0121	3.89
AFM-XGBoost	0.782 ± 0.0086	0.730 ± 0.0047	0.3948 ± 0.0068	0.4805 ± 0.0139	13.97
AFM-XGBoost-FER	0.782 ± 0.0093	0.727 ± 0.0057	0.3961 ± 0.0069	0.4822 ± 0.0135	7.37
AFM-LightGBM	0.787 ± 0.0092	0.736 ± 0.0049	0.3931 ± 0.0067	0.4763 ± 0.0131	6.1
AFM-LightGBM-FER	0.784 ± 0.0091	0.727 ± 0.0052	0.3958 ± 0.0067	0.4814 ± 0.0130	6.9

Table 6. Bridge-Algebra06 evaluation results (mean $\pm$ standard deviation across 5 cross-validation folds).

Bridge-Algebra06	ACC	AUC	RMSE	NLL	TE
DAS3H-LR	0.849 $\pm$ 0.0028	0.790 $\pm$ 0.0015	0.3333 $\pm$ 0.0028	0.3624 $\pm$ 0.0052	85.4
DAS3H-LR-FER	0.848 $\pm$ 0.0028	0.799 $\pm$ 0.0016	0.3333 $\pm$ 0.0028	0.3624 $\pm$ 0.0052	54.8
DAS3H-XGBoost	0.846 $\pm$ 0.0035	0.783 $\pm$ 0.0009	0.3394 $\pm$ 0.0031	0.3760 $\pm$ 0.0052	238.8
DAS3H-XGBoost-FER	0.856 $\pm$ 0.0032	0.777 $\pm$ 0.0015	0.3394 $\pm$ 0.0030	0.3760 $\pm$ 0.0051	205.8
DAS3H-LightGBM	0.845 $\pm$ 0.0036	0.783 $\pm$ 0.0013	0.3387 $\pm$ 0.0030	0.3744 $\pm$ 0.0050	97.6
DAS3H-LightGBM-FER	0.845 $\pm$ 0.0037	0.782 $\pm$ 0.0014	0.3388 $\pm$ 0.0030	0.3747 $\pm$ 0.0050	87.8
PFA-LR	0.841 $\pm$ 0.0032	0.739 $\pm$ 0.0018	0.3479 $\pm$ 0.0029	0.3936 $\pm$ 0.0056	23.5
PFA-LR-FER	0.839 $\pm$ 0.0033	0.751 $\pm$ 0.0018	0.3478 $\pm$ 0.0029	0.3934 $\pm$ 0.0056	14.3
PFA-XGBoost	0.843 $\pm$ 0.0035	0.764 $\pm$ 0.0022	0.3431 $\pm$ 0.0032	0.3828 $\pm$ 0.0058	41.9
PFA-XGBoost-FER	0.843 $\pm$ 0.0035	0.768 $\pm$ 0.0021	0.3431 $\pm$ 0.0032	0.3828 $\pm$ 0.0058	30.9
PFA-LightGBM	0.843 $\pm$ 0.0035	0.769 $\pm$ 0.0019	0.3426 $\pm$ 0.0031	0.3818 $\pm$ 0.0056	26.4
PFA-LightGBM-FER	0.843 $\pm$ 0.0036	0.769 $\pm$ 0.0019	0.3426 $\pm$ 0.0032	0.3818 $\pm$ 0.0056	21.3
AFM-LR	0.837 $\pm$ 0.0038	0.709 $\pm$ 0.0025	0.3555 $\pm$ 0.0028	0.4104 $\pm$ 0.0053	18.8
AFM-LR-FER	0.837 $\pm$ 0.0038	0.710 $\pm$ 0.0025	0.3555 $\pm$ 0.0029	0.4102 $\pm$ 0.0053	14.3
AFM-XGBoost	0.839 $\pm$ 0.0035	0.722 $\pm$ 0.0021	0.3522 $\pm$ 0.0029	0.4034 $\pm$ 0.0053	36.0
AFM-XGBoost-FER	0.839 $\pm$ 0.0036	0.726 $\pm$ 0.0021	0.3521 $\pm$ 0.0029	0.4033 $\pm$ 0.0053	35.5
AFM-LightGBM	0.839 $\pm$ 0.0037	0.724 $\pm$ 0.0023	0.3520 $\pm$ 0.0030	0.4029 $\pm$ 0.0053	18.2
AFM-LightGBM-FER	0.839 $\pm$ 0.0037	0.724 $\pm$ 0.0024	0.3520 $\pm$ 0.0029	0.4029 $\pm$ 0.0053	18.2

Table 7. Assist09 evaluation results (mean $\pm$ standard deviation across 5 cross-validation folds).

Assist09	ACC	AUC	RMSE	NLL	TE
DAS3H-LR	0.700 $\pm$ 0.0056	0.725 $\pm$ 0.0023	0.4395 $\pm$ 0.0026	0.5686 $\pm$ 0.0054	10.95
DAS3H-LR-FER	0.700 $\pm$ 0.0051	0.725 $\pm$ 0.0018	0.4397 $\pm$ 0.0026	0.5687 $\pm$ 0.0053	3.01
DAS3H-XGBoost	0.706 $\pm$ 0.0087	0.710 $\pm$ 0.0034	0.4404 $\pm$ 0.0038	0.5720 $\pm$ 0.0075	31.06
DAS3H-XGBoost-FER	0.707 $\pm$ 0.0078	0.708 $\pm$ 0.0045	0.4414 $\pm$ 0.0037	0.5741 $\pm$ 0.0075	27.17
DAS3H-LightGBM	0.712 $\pm$ 0.0058	0.725 $\pm$ 0.0048	0.4381 $\pm$ 0.0031	0.5672 $\pm$ 0.0062	20.3
DAS3H-LightGBM-FER	0.711 $\pm$ 0.0061	0.721 $\pm$ 0.0045	0.4392 $\pm$ 0.0030	0.5696 $\pm$ 0.0060	10.9
PFA-LR	0.706 $\pm$ 0.0076	0.724 $\pm$ 0.0082	0.4477 $\pm$ 0.0038	0.5879 $\pm$ 0.0077	1.14
PFA-LR-FER	0.690 $\pm$ 0.0074	0.685 $\pm$ 0.0076	0.4479 $\pm$ 0.0036	0.5881 $\pm$ 0.0073	0.82
PFA-XGBoost	0.708 $\pm$ 0.0083	0.720 $\pm$ 0.0093	0.4412 $\pm$ 0.0049	0.5747 $\pm$ 0.0109	5.76
PFA-XGBoost-FER	0.701 $\pm$ 0.0090	0.699 $\pm$ 0.0093	0.4421 $\pm$ 0.0049	0.5765 $\pm$ 0.0103	3.49
PFA-LightGBM	0.711 $\pm$ 0.0082	0.719 $\pm$ 0.0085	0.4382 $\pm$ 0.0043	0.5669 $\pm$ 0.0090	4.6
PFA-LightGBM-FER	0.710 $\pm$ 0.0082	0.715 $\pm$ 0.0078	0.4397 $\pm$ 0.0042	0.5702 $\pm$ 0.0085	7.2
AFM-LR	0.671 $\pm$ 0.0099	0.631 $\pm$ 0.0048	0.4604 $\pm$ 0.0039	0.6129 $\pm$ 0.0081	1.07
AFM-LR-FER	0.671 $\pm$ 0.0096	0.637 $\pm$ 0.0042	0.4601 $\pm$ 0.0039	0.6121 $\pm$ 0.0081	0.80
AFM-XGBoost	0.673 $\pm$ 0.0092	0.639 $\pm$ 0.0050	0.4602 $\pm$ 0.0041	0.6135 $\pm$ 0.0088	4.90
AFM-XGBoost-FER	0.671 $\pm$ 0.0099	0.647 $\pm$ 0.0045	0.4607 $\pm$ 0.0042	0.6141 $\pm$ 0.0091	3.55
AFM-LightGBM	0.674 $\pm$ 0.0087	0.647 $\pm$ 0.0051	0.4591 $\pm$ 0.0040	0.6102 $\pm$ 0.0083	4.1
AFM-LightGBM-FER	0.673 $\pm$ 0.0096	0.642 $\pm$ 0.0049	0.4601 $\pm$ 0.0041	0.6122 $\pm$ 0.0086	3.5

Table 8. Assist17 evaluation results (mean $\pm$ standard deviation across 5 cross-validation folds).

Assist17	ACC	AUC	RMSE	NLL	TE
DAS3H-LR	0.677 $\pm$ 0.0018	0.693 $\pm$ 0.0030	0.4610 $\pm$ 0.0007	0.6131 $\pm$ 0.0014	37.00
DAS3H-LR-FER	0.661 $\pm$ 0.0021	0.678 $\pm$ 0.0029	0.4608 $\pm$ 0.0007	0.6126 $\pm$ 0.0014	15.53
DAS3H-XGBoost	0.685 $\pm$ 0.0039	0.709 $\pm$ 0.0029	0.4537 $\pm$ 0.0013	0.5970 $\pm$ 0.0025	45.67
DAS3H-XGBoost-FER	0.678 $\pm$ 0.0047	0.703 $\pm$ 0.0024	0.4534 $\pm$ 0.0016	0.5964 $\pm$ 0.0032	23.68
DAS3H-LightGBM	0.677 $\pm$ 0.0036	0.708 $\pm$ 0.0037	0.4532 $\pm$ 0.0013	0.5965 $\pm$ 0.0025	57.1
DAS3H-LightGBM-FER	0.674 $\pm$ 0.0045	0.707 $\pm$ 0.0030	0.4537 $\pm$ 0.0016	0.5974 $\pm$ 0.0031	27.2
PFA-LR	0.647 $\pm$ 0.0037	0.620 $\pm$ 0.0053	0.4735 $\pm$ 0.0010	0.6402 $\pm$ 0.0023	6.63
PFA-LR-FER	0.648 $\pm$ 0.0044	0.627 $\pm$ 0.0042	0.4708 $\pm$ 0.0012	0.6342 $\pm$ 0.0024	5.08
PFA-XGBoost	0.659 $\pm$ 0.0023	0.650 $\pm$ 0.0043	0.4654 $\pm$ 0.0010	0.6230 $\pm$ 0.0020	17.63
PFA-XGBoost-FER	0.667 $\pm$ 0.0030	0.662 $\pm$ 0.0037	0.4632 $\pm$ 0.0012	0.6178 $\pm$ 0.0024	9.43
PFA-LightGBM	0.661 $\pm$ 0.0025	0.661 $\pm$ 0.0041	0.4641 $\pm$ 0.0011	0.6201 $\pm$ 0.0022	20.5
PFA-LightGBM-FER	0.660 $\pm$ 0.0035	0.666 $\pm$ 0.0037	0.4632 $\pm$ 0.0013	0.6179 $\pm$ 0.0026	13.5
AFM-LR	0.635 $\pm$ 0.0050	0.589 $\pm$ 0.0040	0.4781 $\pm$ 0.0015	0.6491 $\pm$ 0.0030	7.41
AFM-LR-FER	0.641 $\pm$ 0.0055	0.613 $\pm$ 0.0033	0.4738 $\pm$ 0.0014	0.6401 $\pm$ 0.0029	4.78
AFM-XGBoost	0.643 $\pm$ 0.0041	0.620 $\pm$ 0.0032	0.4732 $\pm$ 0.0014	0.6390 $\pm$ 0.0028	16.38
AFM-XGBoost-FER	0.645 $\pm$ 0.0045	0.630 $\pm$ 0.0031	0.4707 $\pm$ 0.0015	0.6334 $\pm$ 0.0032	8.41
AFM-LightGBM	0.641 $\pm$ 0.0047	0.627 $\pm$ 0.0035	0.4719 $\pm$ 0.0015	0.6360 $\pm$ 0.0031	11.4
AFM-LightGBM-FER	0.642 $\pm$ 0.0047	0.634 $\pm$ 0.0031	0.4705 $\pm$ 0.0016	0.6328 $\pm$ 0.0033	9.3

We additionally computed paired Wilcoxon signed-rank tests on the per-fold AUC values to assess the statistical significance of the main effects (XGBoost vs. logistic regression baseline; FER vs. non-FER variants) for each base model on each dataset. While several comparisons reach statistical significance at $\alpha = 0.05$ (e.g., PFA-LR vs. PFA-XGBoost on both Algebra05 and Bridge-Algebra06), a number of comparisons that show consistent directional differences in mean AUC do not reach this threshold (e.g., AFM-LR vs. AFM-XGBoost on Bridge-Algebra06, $p = 0.0625$; DAS3H-XGBoost vs. DAS3H-XGBoost-FER on Bridge-Algebra06, $p = 0.0625$). This pattern is consistent with a well-documented structural property of the Wilcoxon signed-rank test at small sample sizes: with exactly $n = 5$ paired observations, the test's exact null distribution admits a minimum achievable two-sided p -value of 0.0625 obtained precisely when all five folds agree in the direction of the difference which already exceeds the conventional $\alpha = 0.05$ threshold. Reaching statistical significance under these conditions is therefore constrained by the sample size itself rather than by the absence of a true underlying effect. The Wilcoxon signed-rank test is known to exhibit limited statistical power when the number of paired observations is small. Since our evaluation relies on only five cross-validation folds, the absence of statistical significance should therefore be interpreted cautiously, as it may reflect insufficient statistical power rather than the absence of a practical difference between methods [41,42]. Given this structural constraint and the additional length it would introduce, we report it transparently rather than overstating the strength of the evidence; the standard deviations presented in Tables 5–8 are accordingly offered as the primary, sample-independent indicator of result stability.

The experimental results across the four datasets reveal consistent and complementary behavioral patterns for the three components of the proposed hybrid framework. Regarding the effect of FER-based Q-matrix simplification on execution time, the reductions are substantial and systematic across all configurations. On Algebra05, DAS3H-LR-FER achieves a 66% time reduction (36.00 s $\rightarrow$ 12.07 s) while simultaneously improving AUC from 0.826 to 0.829, and PFA-LR-FER reduces execution time by 74% (15.95 s $\rightarrow$ 4.19 s) with a marginal AUC gain of +0.004.

On Bridge-Algebra06 the largest dataset with 1,817,393 interactions DAS3H-LR-FER reduces training time by 36% (85.4 s $\rightarrow$ 54.8 s) while improving AUC by +0.009 (0.790 $\rightarrow$ 0.799), and PFA-LR-FER achieves a 39% time reduction with an AUC gain of +0.012. On Assist09, DAS3H-LR-FER achieves the most pronounced relative reduction of 72.5% (10.95 s $\rightarrow$ 3.01 s) with a perfectly stable AUC of 0.725, confirming that the $O(I \times K) \rightarrow O(I)$ complexity reduction is particularly effective on datasets with high skill-to-item ratios. On Assist17, DAS3H-LR-FER reduces execution time by 58% (37.00 s $\rightarrow$ 15.53 s), though this comes with a slight AUC decrease of -0.015 , representing the only notable accuracy trade-off observed across all FER configurations. One isolated degradation is also noted for PFA-LR-FER on Assist09, where AUC drops from 0.724 to 0.685, suggesting that PFA's sensitivity to skill granularity is more pronounced on datasets without timestamp information.

These two degradation cases are consistent with the bounded information loss inherent to the FER transformation [17]. For PFA-LR-FER on Assist09, the absence of timestamp information means that PFA relies exclusively on per-skill success and failure counts as its predictive signal; aggregating individual skills into combined skills therefore removes the primary source of discriminative information for this model on this dataset. For DAS3H-LR-FER on Assist17, the relatively high interaction density per learner combined with moderate skill diversity amplifies the cost of skill aggregation, as individual skill trajectories carry more diagnostic value in this configuration. Importantly, the FER-XGBoost combination partially recovers this loss: PFA-XGBoost-FER on Assist17 achieves AUC = 0.662 compared to 0.650 for PFA-XGBoost without FER, confirming that the non-linear correction provided by XGBoost compensates for the granularity reduction introduced by the FER transformation.

Regarding the contribution of XGBoost, its integration improves AUC over the logistic regression baselines for the PFA and AFM models across most datasets, while its effect on DAS3H is more nuanced and dataset-dependent. This distinction reflects a principled difference in the residual non-linear signal available to XGBoost across model families: PFA and AFM, with their simpler linear formulations, leave substantial non-linear patterns uncaptured that XGBoost can exploit; DAS3H, with its richer feature engineering incorporating temporal decay windows and item difficulty, already approximates the predictive ceiling of the shared feature space, leaving limited room for boosting to operate. Specifically, DAS3H-XGBoost underperforms DAS3H-LR on Algebra05 (-0.004 AUC), Bridge-Algebra06 (-0.007 AUC), and Assist09 (-0.015 AUC), while achieving a meaningful gain on Assist17 (+0.016 AUC), where the temporal features carry less discriminative weight. This behavior is consistent with findings reported in our prior work [18], where DAS3H-XGBoost improvements were similarly confined to specific dataset configurations: on Algebra05, PFA-XGBoost gains +0.026 AUC over PFA-LR (0.770 vs. 0.744), and AFM-XGBoost gains +0.007 (0.730 vs. 0.723); on Bridge-Algebra06, PFA-XGBoost improves AUC by +0.025 (0.764 vs. 0.739) and AFM-XGBoost by +0.013 (0.722 vs. 0.709); on Assist17, DAS3H-XGBoost achieves a +0.016 gain (0.709 vs. 0.693). However, XGBoost introduces a significant computational overhead on this sparse, high-dimensional feature space, with execution times multiplied by a factor of 2 to 4 relative to the logistic regression baseline. For instance, DAS3H-XGBoost on Bridge-Algebra06 requires 238.8 s compared to 85.4 s for DAS3H-LR. As shown below, this overhead is specific to XGBoost's handling of sparse one-hot features rather than an inherent property of gradient boosting in general.

To verify that this overhead reflects XGBoost's specific behavior on sparse, high-dimensional one-hot feature matrices rather than gradient boosting in general, we additionally evaluated LightGBM, a histogram-based boosting framework natively optimized for sparse inputs, under the same 5-fold protocol. LightGBM consistently matches or

exceeds XGBoost's predictive performance while requiring substantially less training time: on Bridge-Algebra06, DAS3H-LightGBM achieves an AUC of 0.783 ± 0.0013 (identical to DAS3H-XGBoost) in 97.6 s, less than half of DAS3H-XGBoost's 238.8 s; on Algebra05, PFA-LightGBM improves AUC over PFA-XGBoost (0.775 vs. 0.770) in 6.9 s versus 22.0 s; and on Assist09, AFM-LightGBM achieves the best AUC among all AFM variants (0.647) at less than the training cost of AFM-XGBoost (4.1 s vs. 4.9 s). These results confirm that the computational overhead observed for XGBoost is attributable to its native handling of sparse Q-matrix-derived features rather than to an inherent limitation of gradient boosting, and they position LightGBM as a strong additional baseline that does not alter the central conclusions of this study: FER consistently reduces computational cost across all three boosting/regression backbones, and the relative benefit of boosting over linear models remains concentrated on the PFA and AFM model families, where DAS3H's richer feature set leaves comparatively little residual signal for any boosting method XGBoost or LightGBM to exploit.

The FER-XGBoost combination effectively mitigates this overhead: on Algebra05, DAS3H-XGBoost-FER reduces execution time from 127.2 s to 90.2 s (−29%) while preserving an AUC of 0.821; on Assist17, PFA-XGBoost-FER simultaneously improves AUC from 0.650 to 0.662 (+0.012 AUC) and reduces execution time from 17.63 s to 9.43 s (−47%), representing the most compelling demonstration of the synergy between structural simplification and boosting. Similarly, AFM-XGBoost-FER on Assist17 reduces execution time by 49% (16.38 s → 8.41 s) while improving AUC from 0.620 to 0.630 (+0.010 AUC). Taken together, these results confirm that the FER transformation acts as an algorithmic efficiency multiplier that does not compromise and in many configurations actually enhances the predictive gains introduced by XGBoost, validating the design rationale of the proposed two-stage hybrid algorithm.

Overall, these results indicate that FER is particularly beneficial on datasets where Q-matrix complexity contributes substantially to feature construction cost, yielding significant reductions in execution time while maintaining competitive predictive performance. XGBoost, in turn, provides the largest predictive gains for PFA and AFM, whose simpler linear formulations leave additional non-linear patterns unexploited. For DAS3H, the benefit of XGBoost is generally smaller, as the model already incorporates a rich set of predictive features. The two components are therefore complementary: FER primarily addresses structural Q-matrix complexity, whereas XGBoost captures residual non-linear relationships, resulting in a favorable trade-off between computational efficiency and predictive accuracy.

5. Conclusions & Future Work

This work has proposed and formally analyzed a hybrid algorithm that combines Q-matrix simplification (FER) with gradient boosting (XGBoost) to jointly optimize computational efficiency and predictive accuracy in learner performance prediction. The FER algorithm reduces the feature construction complexity from $O(I \times K)$ to $O(I)$, while XGBoost corrects the residual non-linear patterns of the resulting linear base models. The experimental results across multiple real-world datasets demonstrate that simplifying the Q-matrix substantially reduces execution time without significantly affecting AUC, making classical models more suitable for real-time applications. On the other hand, XGBoost improves predictive performance by capturing complex and non-linear patterns in the data, although at the cost of higher computational demand. When combined, the two approaches offer a balanced trade-off, delivering competitive predictive accuracy while maintaining more efficient execution times. This synergy highlights the potential of hybrid modeling strategies in intelligent tutoring systems, particularly in large-scale environments requiring both responsiveness and robustness.

Beyond the experimental results, the proposed FER-XGBoost framework has practical implications for large-scale Intelligent Tutoring Systems (ITSs). In such environments, learner models must often be updated continuously for thousands of concurrent users. By reducing the structural complexity of the Q-matrix, FER decreases feature generation and model execution costs, enabling faster learner state estimation and recommendation generation. Combined with XGBoost, which maintains competitive predictive performance, the framework can support real-time adaptive learning scenarios where instructional content and learning activities are dynamically recommended according to the learner's predicted needs.

Several directions could extend this work. Exploring more fine-grained strategies for Q-matrix simplification, such as dynamic skill clustering techniques, may further improve the trade-off between speed and accuracy. A systematic comparison with deep and transformer-based knowledge tracing models including DKT, DKVMN, SAKT, AKT, SAINT, and GKT would allow a more comprehensive assessment of the FER-XGBoost framework relative to the current state of the art. Such a study could also explore whether the FER Q-matrix simplification strategy is compatible with attention-based architectures, potentially yielding a unified framework combining structural efficiency, interpretability, and deep predictive modeling. Finally, conducting evaluations in online and real-time environments would help confirm the practical effectiveness of the proposed approach in adaptive learning contexts.

Author Contributions: I.M.: Conceptualization, methodology, software, formal analysis, investigation, resources, data curation, visualization, writing original draft preparation; S.H.: visualization, writing—review and editing; A.O.: data curation, writing—review and editing; A.A.L.: project administration, Validation. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Al-Khwarizmi Program funded by Morocco's Ministry of Education, Ministry of Industry, and the Digital Development Agency (ADD) under Project No. 451/2020 (Smart Learning).

Institutional Review Board Statement: Not applicable, as the study uses publicly available anonymized benchmark datasets.

Informed Consent Statement: Not applicable, as the study uses publicly available anonymized benchmark datasets.

Data Availability Statement: The datasets used in this study are publicly available at the following sources: ASSISTments datasets: <https://sites.google.com/site/assistmentsdata> (accessed on 1 December 2025); KDD Cup 2010 datasets (Algebra05, Bridge-Algebra06): <http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp> (accessed on 1 December 2025). The source code and experimental scripts supporting the findings of this study are openly available at: <https://github.com/theophilegervet/learner-performance-prediction> (accessed on 1 December 2025), <https://github.com/BenoitChoffin/das3h> (accessed on 1 December 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Atkinson, R.C.; Wilson, H.A. Computer-Assisted Instruction. *Science* **1968**, *162*, 73–77. [[CrossRef](#)] [[PubMed](#)]
2. Anderson, J.R.; Boyle, C.F.; Reiser, B.J. Intelligent Tutoring Systems. *Science* **1985**, *228*, 456–462. [[CrossRef](#)] [[PubMed](#)]
3. Nwana, H.S. Intelligent Tutoring Systems: An Overview. *Artif. Intell. Rev.* **1990**, *4*, 251–277. [[CrossRef](#)]
4. Wang, T.H. What Strategies Are Effective for Formative Assessment in an E-learning Environment? *J. Comput. Assist. Learn.* **2007**, *23*, 171–186. [[CrossRef](#)]
5. Shute, V.J. Focus on Formative Feedback. *Rev. Educ. Res.* **2008**, *78*, 153–189. [[CrossRef](#)]
6. Kerman, N.T.; Banihashem, S.K.; Karami, M.; Er, E.; Van Ginkel, S.; Noroozi, O. Online Peer Feedback in Higher Education: A Synthesis of the Literature. *Educ. Inf. Technol.* **2024**, *29*, 763–813. [[CrossRef](#)]

7. Oussous, A.; Menyani, I.; Srifi, M.; Lahcen, A.A.; Kheraz, S.; Benjelloun, F.-Z. An Evaluation of Open Source Adaptive Learning Solutions. *Information* **2023**, *14*, 57. [\[CrossRef\]](#)
8. Pavlik, P.I.; Cen, H.; Koedinger, K.R. Performance Factors Analysis—A New Alternative to Knowledge Tracing. In *Artificial Intelligence in Education; Frontiers in Artificial Intelligence and Applications*; IOS Press: Amsterdam, The Netherlands, 2009. [\[CrossRef\]](#)
9. Cen, H.; Koedinger, K.; Junker, B. Comparing Two IRT Models for Conjunctive Skills. In *Intelligent Tutoring Systems*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5091, pp. 796–798. [\[CrossRef\]](#)
10. Cen, H.; Koedinger, K.; Junker, B. Learning Factors Analysis—A General Method for Cognitive Model Evaluation and Improvement. In *Intelligent Tutoring Systems*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4053, pp. 164–175. [\[CrossRef\]](#)
11. Choffin, B.; Popineau, F.; Bourda, Y.; Vie, J.-J. DAS3H: Modeling Student Learning and Forgetting for Optimally Scheduling Distributed Practice of Skills. *arXiv* **2019**. [\[CrossRef\]](#)
12. Wilson, M.; De Boeck, P. Descriptive and Explanatory Item Response Models. In *Explanatory Item Response Models*; Springer: New York, NY, USA, 2004; pp. 43–74. [\[CrossRef\]](#)
13. Corbett, A.T.; Anderson, J.R. Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Model. User-Adapt. Interact.* **1995**, *4*, 253–278. [\[CrossRef\]](#)
14. Shen, S.; Liu, Q.; Huang, Z.; Zheng, Y.; Yin, M.; Wang, M.; Chen, E. A Survey of Knowledge Tracing: Models, Variants, and Applications. *IEEE Trans. Learn. Technol.* **2024**, *17*, 1858–1879. [\[CrossRef\]](#)
15. Abdelrahman, G.; Wang, Q.; Nunes, B. Knowledge Tracing: A Survey. *ACM Comput. Surv.* **2023**, *55*, 224. [\[CrossRef\]](#)
16. Tatsuoaka, K.K. Rule space: An approach for dealing with misconceptions based on item response theory. *J. Educ. Meas.* **1983**, *20*, 345–354. [\[CrossRef\]](#)
17. Menyani, I.; Oussous, A.; Lahcen, A.A. Fast E-Learning Recommendation: Enhancing Model Efficiency with Q-Matrix Complexity Reduction. *Int. J. Adv. Comput. Sci. Appl.* **2026**, *17*, 65–74. [\[CrossRef\]](#)
18. Hakkal, S.; Lahcen, A.A. XGBoost to Enhance Learner Performance Prediction. *Comput. Educ. Artif. Intell.* **2024**, *7*, 100254. [\[CrossRef\]](#)
19. Jebbari, M.; Cherradi, B.; Hamida, S.; Ouassil, M.A.; El Harrouti, T.; Raihani, A. Enhancing Learner Performance Prediction on Online Platforms Using Machine Learning Algorithms. *Indones. J. Electr. Eng. Comput. Sci.* **2024**, *35*, 343. [\[CrossRef\]](#)
20. Gao, G.; Marwan, S.; Price, T.W. Early Performance Prediction Using Interpretable Patterns in Programming Process Data. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*; Association for Computing Machinery: New York, NY, USA, 2021; pp. 342–348. [\[CrossRef\]](#)
21. Albreiki, B.; Zaki, N.; Alashwal, H. A Systematic Literature Review of Student' Performance Prediction Using Machine Learning Techniques. *Educ. Sci.* **2021**, *11*, 552. [\[CrossRef\]](#)
22. Pandey, S.; Srivastava, J. RKT: Relation-Aware Self-Attention for Knowledge Tracing. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1205–1214. [\[CrossRef\]](#)
23. Abdelrahman, G.; Wang, Q. Knowledge Tracing with Sequential Key-Value Memory Networks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 175–184. [\[CrossRef\]](#)
24. Gong, Y.; Beck, J.E.; Heffernan, N.T. Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures. In *Intelligent Tutoring Systems*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6094, pp. 35–44. [\[CrossRef\]](#)
25. Gong, Y.; Beck, J.E.; Heffernan, N.T. How to Construct More Accurate Student Models: Comparing and Optimizing Knowledge Tracing and Performance Factor Analysis. *Int. J. Artif. Intell. Educ.* **2011**, *21*, 27–46. [\[CrossRef\]](#)
26. Effenberger, T.; Pelánek, R.; Čechák, J. Exploration of the Robustness and Generalizability of the Additive Factors Model. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 472–479. [\[CrossRef\]](#)
27. Asselman, A.; Khaldi, M.; Aammou, S. Enhancing the Prediction of Student Performance Based on the Machine Learning XGBoost Algorithm. *Interact. Learn. Environ.* **2023**, *31*, 3360–3379. [\[CrossRef\]](#)
28. Al-Tameemi, G.; Xue, J.; Ali, I.H.; Ajit, S. A Hybrid Machine Learning Approach for Predicting Student Performance Using Multi-Class Educational Datasets. *Procedia Comput. Sci.* **2024**, *238*, 888–895. [\[CrossRef\]](#)
29. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 785–794. [\[CrossRef\]](#)
30. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2017.

31. Piech, C.; Bassen, J.; Huang, J.; Ganguli, S.; Sahami, M.; Guibas, L.J.; Sohl-Dickstein, J. Deep Knowledge Tracing. In *Proceedings of the 29th International Conference on Neural Information Processing Systems*; MIT Press: Cambridge, CA, USA, 2015.
32. Zhang, J.; Shi, X.; King, I.; Yeung, D.-Y. Dynamic Key-Value Memory Networks for Knowledge Tracing. In *Proceedings of the 26th International Conference on World Wide Web*; International World Wide Web Conferences Steering Committee: Perth, Australia, 2017; pp. 765–774.
33. Pandey, S.; Karypis, G. A Self-Attentive Model for Knowledge Tracing. *arXiv* **2019**. [[CrossRef](#)]
34. Ghosh, A.; Heffernan, N.; Lan, A.S. Context-Aware Attentive Knowledge Tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; ACM: New York, NY, USA, 2020; pp. 2330–2339.
35. Choi, Y.; Lee, Y.; Cho, J.; Baek, J.; Kim, B.; Cha, Y.; Shin, D.; Bae, C.; Heo, J. Towards an Appropriate Query, Key, and Value Computation for Knowledge Tracing. In *Proceedings of the Seventh ACM Conference on Learning @ Scale, Virtual Event USA*; ACM: New York, NY, USA, 2020; pp. 341–344. [[CrossRef](#)]
36. Nakagawa, H.; Iwasawa, Y.; Matsuo, Y. Graph-Based Knowledge Tracing: Modeling Student Proficiency Using Graph Neural Network. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*; ACM: New York, NY, USA, 2019; pp. 156–163.
37. Vie, J.-J.; Kashima, H. Knowledge Tracing Machines: Factorization Machines for Knowledge Tracing. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 750–757. [[CrossRef](#)]
38. Koedinger, K.R.; Baker, R.S.; Cunningham, K.; Skogsholm, A.; Leber, B.; Stamper, J. A Data Repository for the EDM Community: The PSLC DataShop. *Handb. Educ. Data Min.* **2010**, *43*, 43–56.
39. Feng, M.; Heffernan, N.; Koedinger, K. Addressing the Assessment Challenge with an Online System That Tutors as It Assesses. *User Model. User-Adapt. Interact.* **2009**, *19*, 243–266. [[CrossRef](#)]
40. Gervet, T.; Koedinger, K.; Schneider, J.; Mitchell, T. When Is Deep Learning the Best Approach to Knowledge Tracing? *J. Educ. Data Min.* **2020**, *12*, 31–54. [[CrossRef](#)]
41. Demsar, J.; Demsar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
42. NCSS Statistical Software. *ASS Sample Size Software*; NCSS 2024; Wilcoxon Signed-Rank Test; NCSS: Silver Spring, MD, USA, 2024.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.