

Fast E-Learning Recommendation: Enhancing Model Efficiency with Q-Matrix Complexity Reduction

Ismail MENYANI, Ahmed OUSSOUS, Ayoub AIT LAHCEN

National School of Applied Sciences-The Engineering Sciences Laboratory, Ibn Tofail University, Kenitra, Morocco

Abstract—Intelligent tutoring systems generate a large volume of data, which becomes particularly valuable when effectively leveraged for learner performance prediction in adaptive learning environments. In this context, the speed and predictive accuracy of machine learning models are crucial, as they determine the system’s ability to deliver timely and relevant insights and support responsive, personalized instruction. Enhancing model speed not only increases tutoring efficiency but also improves the adaptability of educational systems to learners’ needs. This study introduces an approach aimed at improving the execution time of three logistic regression-based models widely used for learner performance prediction: DAS3H (*Item Difficulty, Student Ability, Skill, and Student Skill Practice History*), AFM (*Additive Factor Model*), and PFA (*Performance Factor Analysis*). The proposed optimization reduces the complexity of the Q-matrix that links each item to its required knowledge components by simplifying its structure while preserving pedagogical relevance. An empirical evaluation was conducted on four real-world datasets collected from online tutoring platforms. The results demonstrate that the proposed approach, called *Fast E-learning Recommendation (FER)*, significantly improves the execution speed of the three models while maintaining comparable predictive performance across datasets.

Keywords—Learner performance prediction; adaptive learning; complexity; knowledge components; Q-matrix; machine learning; DAS3H; PFA; AFM; IRT

LIST OF ACRONYMS

IRT	Item Response Theory
PFA	Performance Factor Analysis
AFM	Additive Factor Model
DAS3H	Dynamic Assessment System for Holistic Heuristic
ITS	Intelligent Tutoring System

I. INTRODUCTION

Computer-Assisted Instruction (CAI), also known as Computer-Based Learning, marks the early attempts to integrate technology into education. Initiated in the 1950s with Skinner’s pioneering work on “linear programs”, this approach aimed to automate teaching through standardized sequences. These early systems, mostly textual, offered structured learning but lacked any form of interactivity or adaptation to learners’ specific needs.

The concept of Intelligent Tutoring Systems (ITS), introduced by John R. Anderson and C. Franklin Boyle in 1985 [1], evolved from generative CAI systems [2]. Unlike early CAI, ITS integrates pedagogical content that evolves based on students’ competencies and learning progress. For an ITS to be truly effective, it must continuously and rapidly assess learners’

mastery levels in order to adapt educational content in real time [3,4]. This ability to provide fast and personalized assessments is essential for tracking learner progress and delivering targeted recommendations [5]. Over the past decade, ITS has moved into practical deployment in both educational and industrial sectors. These systems are designed to personalize learning based on learners’ levels, preferences, and pace. For instance, in subjects such as mathematics and physics, systems like ALEKS (Assessment and Learning in Knowledge Spaces) automatically adapt exercises based on mastered skills [6]. In the field of programming, RoboMission introduces learners to algorithmic logic through interactive missions [7]. Other platforms, such as Knewton, use adaptive learning algorithms to dynamically adjust content according to student progress. These systems aim to enhance learning efficiency by delivering tailored exercises at the right time with immediate feedback [8].

In the era of Big Data, educational systems generate vast amounts of data on learner interactions. These data enable deep analysis of learner behavior, a better understanding of their knowledge levels and preferences, thereby improving their learning experience. However, assessing mastery of specific skills remains a challenge. To address this, ITS integrates techniques for tracking student knowledge and predicting performance. Among the most widely used models are: Performance Factor Analysis (PFA) [9], Additive Factor Model (AFM) [10,11], and Difficulty, Student Ability, Skill, and Student Skill practice History (DAS3H) [12]. Others include Item Response Theory (IRT) [13] and Bayesian Knowledge Tracing (BKT) [14].

Although these models are effective for estimating learners’ mastery levels [12], they exhibit computational limitations, especially in real-time environments. Their increasing complexity poses a major challenge for large-scale deployment. Furthermore, research efforts tend to focus on predictive performance improvement, often at the expense of algorithmic simplicity. This trend results in increasingly sophisticated but also more computationally heavy, slower, and resource-intensive models.

In this context, a key component of these models is the learning matrix, built from a Q-matrix [15], which links each item to the knowledge components it engages. This binary matrix (with only 0 and 1) enables the model to associate learner performance with specific skills, directly influencing both prediction quality and computation speed. In these models’ Q-matrix structure, an item may simultaneously involve several knowledge components, and different items often draw on the same components. The complexity of the Q-matrix increases with the number of 1 it contains: the more such associations, the

denser the matrix becomes, increasing model complexity and slowing computation. Therefore, the structure and density of the Q-matrix play a decisive role in the performance and efficiency of models such as PFA, AFM, or DAS3H.

To overcome the computational complexity associated with Q-matrix-based models, this study proposes a novel approach called Fast E-learning Recommendation (FER). Unlike traditional approaches that focus mainly on maximizing predictive performance, FER is designed to optimize execution speed while maintaining a comparable level of accuracy to existing models. The method involves transforming the Q-matrix by grouping the skills associated with each item into a combined skill. This simplification reduces matrix density, significantly speeds up the learning process, and retains the ability to adapt to learner profiles.

This work falls within the scope of student performance prediction based on data collected from intelligent tutoring systems. The main objective of this research is to propose an approach that simultaneously improves the predictive accuracy of models while reducing their computational cost. To achieve this, a novel method called Fast E-learning Recommendation (FER) is introduced, applied to classical models such as PFA, AFM, and DAS3H. The method is based on a simplification of the Q-matrix through skill combination, effectively reducing its density without significant performance loss. This method is extended to derive enhanced versions of the models DAS3H-FER, PFA-FER, and AFM-FER which achieve a substantial reduction in computation time while maintaining, and in some cases improving upon, the predictive performance of the original models. An empirical evaluation conducted on four real-world datasets confirms the improvements in both accuracy and speed brought by the proposed approach.

This study is structured as follows: Section II introduces related work on learner performance modeling, with a focus on logistic regression-based models. Section III details the proposed approach. Section IV outlines the adopted methodology, including data collection and the experimental protocol. Section V presents and analyzes the obtained results. Finally, Section VI concludes and discusses future research directions.

II. RELATED WORK

The objective of this research is to reduce the execution time of learner performance prediction models while maintaining a predictive accuracy comparable to that of traditional logistic regression models. To achieve this, the Q-matrix is leveraged as an essential component in learner skill assessment by simplifying its structure through sparsity reduction.

Several recent studies have addressed the issue of computational complexity in knowledge tracing models. For instance, [16] proposed the Sequence-Flexible Knowledge Tracing (SFKT) model, which employs a Transformer-based encoder to efficiently handle learning sequences of variable lengths. While this model maintains constant complexity for long sequences and offers strong predictive performance, it primarily focuses on adaptive sequence processing rather than explicit execution time reduction. This distinction constitutes a fundamental difference from the proposed approach, which

directly targets computational efficiency through structural simplification.

In another contribution, [17] integrated XGBoost [18] with classical logistic regression models such as Item Response Theory [13], Performance Factor Analysis (PFA), and DAS3H, aiming to enhance predictive accuracy. Although this hybrid approach achieved high AUC scores, the improvement in accuracy came at the cost of increased computational complexity. Specifically, the use of 1000 iterations and a low learning rate (0.1) significantly increases both training and inference time. In contrast, the proposed method seeks to preserve or even improve predictive performance while substantially reducing execution time a trade-off not addressed in that study.

Furthermore, in the broader field of matrix optimization, the theoretical work by [19] on fast multiplication of sparse matrices offers valuable insights. Their algorithm improves upon the classical $O(mn)$ complexity by introducing a more efficient solution that performs in $O(m^{0.7}n^{1.2} + n^{2+o(1)})$ operations for matrices with m non-zero elements. While this work remains theoretical and is not directly applied to educational models, it supports the hypothesis that reducing matrix density such as that of the Q-matrix can lower computational costs without compromising the integrity of the encoded information. In the proposed approach, the Q-matrix is treated as a sparse structure, whose simplification through skill aggregation leads to substantial time savings.

Finally, whereas most prior studies focus either on boosting predictive accuracy via model complexity (e.g., ensemble models such as XGBoost) or on architectural innovation (e.g., SFKT), few have systematically explored the trade-off between predictive accuracy and execution speed through transformation of knowledge representation. This gap is precisely what the Fast E-learning Recommendation (FER) method addresses, by restructuring the skill representation space. The resulting models (DAS3H-FER, PFA-FER, and AFM-FER) demonstrate improved robustness and efficiency across multiple real-world datasets.

A. Predicting Learners' Performance Using Logistic Regression

Logistic regression is a widely used method in the field of Educational Data Mining (EDM) due to its reliability in predicting learners' performance. Typically, the data used in this context consists of a set of learners answering a series of questions. For each interaction x_t , the model has access to the learner's identifier, the question in focus q_t , as well as an indication of the learner's success or failure, denoted as a_t : 0 if the learner fails and 1 if they succeed. When the algorithm incorporates a sequential or temporal dimension, it can also account for the position of the interaction within the learner's sequence of interactions or simply the time at which it occurred. The sequence of a learner's past interactions is denoted with the tutoring system as $x_{1:t} = (x_1, \dots, x_t)$, where x_t is a tuple composed of q_t and a_t [20].

Moreover, each interaction may involve one or more Knowledge Components (KCs) associated with the given question. For models that consider learning and forgetting, such

as DAS3H [12], the timestamp feature (TS_t) is included to track these dynamics effectively.

General reformulation of the model based on logistic regression

- S learner;
- k knowledge component (KC);
- q_t question at time t ;
- TS_t Timestamp at time t ;
- $a_t \in \{0, 1\}$ correctness at time t ;
- $x_{1:t}$ all data up to time t (interactions);
- $x_{s,1:t}$ all data of learner s up to time t (learner interactions);
- $\delta(\cdot)$ one hot encoding of feature
- σ the logistic function $\sigma(x) = 1/(1 + e^x)$.

The logistic regression prediction for a student (S) to correctly answer question $q_{s,t+1}$ at time t is mathematically represented as follows:

$$P(a_{s,t+1} = 1 | q_{s,t+1}, x_{s,t+1}) = \sigma(W^T \varphi(q_{s,t+1}, x_{s,t+1}))$$

- W^T a trainable weight vector;
- $\varphi = (\varphi_1, \dots, \varphi_d)$, a features vector.

B. IRT Model

Given the diversity of feature vectors φ that have led to various models in the literature, the first among them is the Item Response Theory (IRT) [21]. The IRT model uses latent parameters to simulate learners' ability levels, denoted as α_s , which are estimated based on the learner's responses to a set of questions, along with the difficulty of the question $\delta q_{s,t+1}$, quantified through an analysis of learners' responses to that item. The IRT model assumes that the learner's ability level is static and does not change over time. Despite its simplicity, the IRT model is robust and reliable, often achieving better predictive performance than more complex models. However, the present study does not address the IRT model, as it does not rely on Knowledge Components (KCs). Instead, the proposed approach focuses on datasets in which items involve multiple Knowledge Components.

The IRT model predicts the probability that a learner s will answer the question $q_{s,t+1}$ correctly at time t as follows:

$$P(a_{s,t+1} = 1 | q_{s,t+1}, x_{s,t+1}) = \rho(\alpha_s - \delta q_{s,t+1})$$

C. AFM Model

In the Additive Factor Model (AFM) [10,11], the probability that a learner answers an item correctly depends on their initial ability level α_s and the ease associated with each knowledge component (KC) required by item $q_{s,t+1}$. However, the learner's practice of these KCs also influences this probability: a positive γ_k coefficient indicates that the probability of a correct response increases as the learner practices the KC. Conversely, a negative γ_k suggests that this probability decreases with practice. The

greater the absolute value of γ_k , the more significant the impact of practice on knowledge component k .

$$P(a_{s,t+1} = 1 | q_{s,t+1}, x_{s,t+1}) = \sigma(\alpha_s + \sum_{k \in KC(q_{s,t+1})} \beta_k + \gamma_k a_{s,k})$$

where,

- k identifier of the knowledge component (KC);
- β_k latent easiness parameter of KC k ;
- $a_{s,k}$ number of attempts made by learner s on KC k before this attempt;
- γ_k coefficient associated with this past attempt counter, which can be interpreted as the learning rate of KC k .

D. PFA Model

The Performance Factors Analysis (PFA) model [9] is an extension of the AFM model. It breaks down past practice into two distinct counters: successes and failures for each knowledge component (KC) involved in the item, rather than relying solely on simple attempt counters. The mathematical formulation of the model is given as follows:

$$P(a_{s,t+1} = 1 | q_{s,t+1}, x_{s,t+1}) = \sigma\left(\sum_{k \in KC(q_{s,t+1})} \beta_k + \gamma_k c_{s,k} + \rho_k f_{s,k}\right)$$

where,

- $c_{s,k}$ the number of correct responses by learner S on KC k before this attempt;
- $f_{s,k}$ the number of incorrect responses by learner S on KC k before this attempt;
- γ_k and ρ_k the coefficients estimated by the model and associated with these counters;
- β_k Difficulty of KC k .

E. DAS3H Model

The DAS3H model [12] combines concepts from the PFA and IRT models to predict whether a learner will correctly answer ($a_{s,t+1}$) an item ($q_{s,t+1}$) associated with one or more knowledge components (KCs). It incorporates four key factors: item difficulty, student ability, knowledge components, and the practice history of the components. Inspired by the DASH model [22], DAS3H accounts for the effects of forgetting and practice by leveraging the temporal structuring of past experiences, thereby modeling the learning and forgetting curves for each KC. By introducing counters for correct responses and attempts within a given time window, this predictive and interpretable model captures the dynamics of learning and forgetting based on the learner's performance history.

In its formulation (with a dimension of $d = 0$ for continuous vector representations), the quadratic term of the KTM disappears, and DAS3H is expressed as follows:

$$\begin{aligned}
 P(a_{s,t+1} = 1 | q_{s,t+1}, x_{s,t+1}) &= \sigma \left(\alpha_k - \gamma_{q_{s,t+1}} \right. \\
 &+ \sum_{k \in KC(q_{s,t+1})} \beta_k \\
 &+ \sum_{k \in KC(q_{s,t+1})} \sum_{w=0}^{w-1} \theta_{k,2w+1} \log(1 + c_{s,k,w}) \\
 &\left. - \theta_{k,2w+2} \log(1 + a_{s,k,w}) \right)
 \end{aligned}$$

where,

- w , indexing a set of W time windows of increasing size;
- $\theta \in \mathbb{R}^{k+2w}$, the matrix of coefficients associated with the counters for successes and attempts. These coefficients describe the learning and forgetting curves of a KC k ;
- $c_{s,k,w}$, the number of correct responses by learner S on item $q_{s,t+1}$ within the time window w ;
- $a_{s,k,w}$, the number of attempts by learner S on item $q_{s,t+1}$ within the time window w .

For higher-dimensional vector embeddings ($d > 0$), the formulation of the DAS3H model becomes more complex. The function σ is replaced by the probit function ϕ [23], which corresponds to the cumulative distribution function of a standard normal random variable $\mathcal{N}(0, 1)$. When implementing the model within the framework of KTMs, it is crucial to define the variables to be encoded in the sparse vector xx . In this context, the learner s , the item $q_{s,t+1}$, the knowledge components (KCs)

$k \in CC(q_{s,t+1})$, as well as the successes $c_{s,k,w}$ and attempts $a_{s,k,w}$ for each time window w are considered.

The DAS3H model is then expressed as follows:

$$\begin{aligned}
 P(a_{s,t+1} = 1 | q_{s,t+1}, x_{s,t+1}) &= \phi \left(\mu + \sum_{i=1}^N \omega_i x_{s,q_{s,t+1},t}^i \right. \\
 &+ \underbrace{\sum_{1 \leq i \leq l \leq N} x_{s,q_{s,t+1},t}^i x_{s,q_{s,t+1},t}^l}_{\text{B}} < \vartheta_i, \vartheta_l > \left. \right)
 \end{aligned}$$

where,

- ϕ probit function;
- μ global bias of the model;
- $x_{s,j,t}^i$ value of feature iii in the model for learner s answering question $q_{s,t+1}$ at time t ;
- ω_i real-valued coefficient estimated for this feature;
- ϑ_i continuous vector representation of this feature;
- N number of features considered.

The second summation includes the pairwise interactions between the continuous vector representations of each variable considered by DAS3H. This second summation, B, enhances DAS3H as it allows modeling the interactions between each pair of variables. This can, for example, be useful if certain learners struggle more with specific KCs than others (see Table I).

TABLE I. FEATURES CONSIDERED BY EACH LEARNER MODEL [24]

Φ	Learner	items	Skills (KCs)	wins	fails	Attempts	Time windows	Execution time
DAS3H	✓	✓	✓	✓		✓	CC	Low
PFA	✓	✓	✓	✓	✓		\emptyset	Moderate
AFM	✓		✓			✓	\emptyset	Medium

III. PROPOSED APPROACH FOR FAST E-LEARNING RECOMMENDATION (FER)

Classical knowledge tracing models such as PFA, AFM, and DAS3H have proven highly effective in predicting learners' mastery levels [9,14]. However, their large-scale deployment faces significant computational challenges, primarily due to the increasing complexity of the Q-matrix on which they rely. This matrix, initially introduced by Tatsuoka (1983) and later adopted in various works [25,26], defines the relationship between each item and the set of skills required to answer it.

In many cases, the Q-matrix is sparse, meaning it contains relatively few 1s compared to its total size. Nonetheless, even in such cases, its density can lead to a significant computational load. For instance, in the *robomission19* dataset, the Q-matrix contains 146,523 rows and 7,320 columns. Such a structure results in a computational cost proportional to the number of

non-zero entries, as shown by [19] in their work on sparse matrix multiplication.

To mitigate this complexity, a method based on the transformation of the Q-matrix is proposed, which consists of grouping the skills required by certain items into combined skills. This idea is indirectly inspired by matrix factorization and representation compression techniques. The objective is to reduce the effective density of the matrix without altering the underlying cognitive structure.

A. Transformation Applied to the Q-Matrix to Reduce the Number of Knowledge Components (KC)

As previously explained, the Q-matrix $Q \in \{0,1\}^{I \times K}$ associates each item $i \in \{1, \dots, I\}$ with a set of $k \in \{1, \dots, K\}$. Here, I represents the set of items, and K the set of skills. The element $q_{ik} = 1$ indicates that item i requires skill k , and $q_{ik} = 0$ otherwise [15,26]. This matrix is typically sparse, meaning that most of its entries are zero.

In knowledge tracing models (e.g., AFM, DAS3H), the computational cost is directly influenced by the number of non-zero elements in the Q-matrix, i.e., the total number of skills engaged across all items [19]. To formalize this idea, we introduce the notation Complexity (Q) (used here as an illustrative measure), defined as follows:

$$\text{Complexity}(Q) =: \sum_{i=1}^I \sum_{k=1}^K q_{ik}$$

This expression represents the total number of ones in the Q-matrix, indicating item–skill associations. It is used in this context as an approximate measure of the computational burden associated with Q-matrix density.

1) *Objective*: The goal of the proposed transformation is to reduce the complexity of the Q-matrix by decreasing the total number of ones it contains, while preserving the essential relationships between items and skills.

2) *Proposed transformation*: Suppose an item i is associated with a set of skills $\{k_1, k_2, \dots, k_{n_j}\}$. This set is replaced by a new combined skill, denoted as c_j , defined by: $c_j = k_1 \wedge k_2 \wedge \dots \wedge k_{n_j}$

In other words, a new skill representing the logical conjunction is created for the initial skills. The updated Q-matrix Q' is then modified according to the following rule:

$$q'_{ik} = \begin{cases} 1, & \text{if } k = c_j \\ 0, & \text{if } k \in \{k_1, k_2, \dots, k_{n_j}\} \\ q_{ik}, & \text{otherwise} \end{cases}$$

This operation reduces the number of ones in the matrix while retaining the essential cognitive requirements. The combined skill c_j encapsulates all the original skill requirements. Thus, a learner who masters the combined skill is assumed to have mastered all its component skills.

3) *Illustrative example*: Consider an initial Q-matrix, where item 1 requires both skill KC_1 and skill KC_3 . After transformation, these two skills are replaced by a new combined skill $KC_{1,3}$ and the corresponding row in the matrix changes from two 1s to a single 1:

item	KC_1	KC_2	KC_3	$KC_{2,3}$
1	0	0	0	1

This simplification reduces the overall density of the Q-matrix. The principle is similar to that of dimensionality reduction or skill clustering, which is common in probabilistic cognitive modeling [5,10], and it aligns with recommendations in the field of sparse matrix optimization for large-scale computation [19].

Although FER shares superficial resemblance with several established techniques, it differs from them in both motivation and design. Feature hashing [27] and item-level feature collapsing are general-purpose dimensionality reduction strategies applied to input representations, typically without regard for the underlying cognitive structure of the data. In

contrast, FER operates directly on the Q-matrix a pedagogically grounded structure and its transformation is driven by the semantics of knowledge components rather than by hash functions or arbitrary feature groupings. Skill clustering methods [28] such as those based on learner response co-occurrence or inter-item similarity aim to discover latent groupings of knowledge components across the entire dataset, which requires an additional unsupervised learning step and may alter the interpretability of the model. FER, by contrast, performs a deterministic, item-level aggregation that preserves the original item–skill associations in a compressed form, without introducing any learned latent structure. Dimensionality reduction techniques [29] such as PCA or SVD operate in a continuous, dense embedding space and are designed to approximate a dense matrix; they are not tailored to sparse binary structures such as the Q-matrix. FER explicitly targets such sparse binary matrices and preserves their discrete, interpretable nature, making it semantically compatible with cognitive diagnostic frameworks. Finally, unlike item-specific latent feature models [30] (e.g., within Factorization Machines or Knowledge Tracing Models), which learn continuous vector representations per item through iterative optimization, FER requires no additional training overhead. Its computational simplicity and transparency make it particularly well-suited for real-time adaptive learning environments where both speed and pedagogical interpretability are critical requirements.

IV. METHODOLOGY

This section introduces the experiment settings by describing the datasets selected to perform this study (see Table II), then the experiment methodology and the evaluation metrics.

A. Datasets

This study is based on interaction data between learners and items, where each response is classified as correct or incorrect. The selected datasets include both temporal information and indications of the skills (KCs) involved in each item. The presence of multiple skills per item was essential, as the proposed method relies precisely on this characteristic. These datasets are frequently used in the literature, enabling a relevant comparative evaluation of the predictive performance and execution time of the DAS3H, PFA, and AFM models.

ASSISTments 2009-2010 and 2012-2013: ASSISTments is an intelligent tutoring platform for mathematics that provides personalized support to students through exercises accompanied by hints and feedback. The 2009–2010 (assist09) and 2016–2017 (assist17) versions were used, which include items involving multiple skills [31,32].

Algebra I 2005-2006 and Bridge to Algebra 2006-2007: These data, derived from Carnegie Learning's cognitive tutors, come from the 2005–2006 to 2006–2007 school years. It was used in the KDD Cup challenge to develop analytical models based on real-world data from algebra learning [33].

RoboMission 2019: RoboMission is a programming learning system based on space missions, offering progressive tasks structured into levels and sub-levels. Data from December 2019 (Robomission19) were used, which allows for the evaluation of learners' progressive mastery of concepts [12].

TABLE II. STATISTICS ON THE DATASETS USED AFTER PREPROCESSING

Dataset	Learners	Items	KCs	Interactions	KCs per item
assist17	1708	3162	102	934638	1,225
bridge06	1130	129263	493	1817393	1,014
algebra05	567	173113	112	606983	1,363
robomission19	7066	85	15	146523	7,412

TABLE III. STATISTICS ON THE DATASETS USED AFTER COMBINING KNOWLEDGE COMPONENTS.

Dataset	Learners	Items	KCs	Interactions	KCs per item
assist17	1708	3162	393	934638	1
bridge06	1130	129263	550	1817393	1
algebra05	567	173113	216	606983	1
robomission19	7066	85	54	146523	1

Significant variation is observed across the datasets in terms of the number of knowledge components (KCs), items, learners, and temporal features (timestamps). These timestamps are available in all datasets and are utilized by the three models: DAS3H, PFA, and AFM.

After transformation, an increase in the total number of knowledge components is observed across all datasets, reflecting the creation of new combined components. This operation simplifies the structure of the Q-matrix while capturing complex associations between skills. Moreover, the average number of KCs per item is reduced to 1, indicating that each item is now associated with a single combined component, corresponding to the total number of items in each dataset (see Table III).

B. Experiment

1) Data preprocessing (data preparation, data encoding):

The preprocessing of data is a fundamental phase in any machine learning workflow, particularly because the log data collected from students is not immediately suitable for model input. To prevent biases and inconsistencies, it is essential to clean and convert the raw data into a structured format. This preparation involves applying a variety of methods aimed at enhancing the data's quality, precision, and consistency, which in turn contributes to the overall performance of the model. Once the necessary data has been gathered, a detailed analysis of its structure and attributes is performed. This involves managing missing values specifically by removing entries lacking skill tags, renaming fields, extracting timestamps, filtering out continuous outcomes, selecting user sequences with a sufficient number of interactions, and organizing the data chronologically and by user. These preprocessing procedures were consistently applied to the ASSISTments, KDD, and Robomission datasets.

Subsequently, the Q-matrix is constructed as part of the data preprocessing phase. It plays a central role in many cognitive diagnostic models and educational data analysis procedures. This binary matrix captures the relationships between items (questions) and knowledge components (skills). Each row

corresponds to an item, each column to a skill, and an entry of 1 indicates that the item engages the corresponding skill, while a value of 0 signifies no association.

Once the items and skills are identified in the datasets, the Q-matrix is initialized as a zero matrix. The relevant associations are then encoded as 1. To optimize the storage of this mostly sparse structure, the Q-matrix is represented as a sparse matrix, which stores only non-zero entries and their positions. This significantly reduces memory usage, particularly when the number of skills is much smaller than the total number of items.

To further optimize this matrix, the Fast E-learning Recommendation (FER) approach is applied, which restructures the Q-matrix so that each item is associated with only one aggregated skill. This simplification reduces the matrix density, enhances computational efficiency, and preserves the pedagogical coherence of the model. Then, the resulting sparse features matrix is saved as (.npz) to be used for the performance prediction.

2) *Preservation of information*: The transformation of the Q-matrix, by combining multiple skills into a single composite skill, does not lead to significant information loss, as the new skill encapsulates the original skills in a simplified manner. To better illustrate this idea, consider an item i requiring n skills: $CC_1, CC_2, CC_3, \dots, CC_n$. The new combined skill CC_k replaces these individual skills. Thus, if a student masters the combined skill CC_k it implies mastery of the combination of these skills, though not necessarily each individual skill. This approach entails only a slight loss of information, as the combined skill captures the essence of the original skill set, without reflecting all their individual nuances.

Moreover, it is worth noting that the datasets used contain a large number of distinct items that often involve the same skills or combinations already seen in other items. This redundancy further mitigates the potential information loss by ensuring consistency and recurrence in the appearance of knowledge components, even after transformation.

3) *Implementation*: This study aimed to reduce the training time of logistic regression models by simplifying the complexity of the Q-matrix while grouping the knowledge components associated with each item. The research objectives included evaluating the predictive performance of these models on four real-world educational datasets. To ensure a fair comparison between the models, all were implemented within the KTM framework [34] and evaluated using three different embedding dimensions [35].

The models were developed in Python on a Windows 11 operating system with 16 GB of RAM and an Intel Core i7 11th-generation processor. As in Vie et al. (2019), the *pywFM* library was used as a wrapper for *libfm* [23] for models with $d > 0$. In these cases, the number of iterations for the Gibbs MCMC sampler was set to 500. When $d = 0$, the *scikit-learn* library (Pedregosa et al., 2011) and its implementation of logistic regression with L_2 regularization were used. The SAGA algorithm [36] was selected for optimization, with the regularization parameter set to $C = 1$.

The following sections present a discussion and analysis of the experimental results, along with a perspective on the effectiveness of the proposed approach.

V. RESULTS AND DISCUSSION

In this section, the performance results are presented for each dataset trained using the new approach proposed in this study. These results are compared to those obtained with the other three models. The metrics used are the Area Under the Curve (AUC)

and Execution Time (TE), which are reported for both models and datasets.

The analysis of the results highlights the effectiveness of the FER versions of the models (FER-DAS3H, FER-PFA, FER-AFM) in reducing execution time while maintaining acceptable performance levels across the studied datasets. This reduction in time is particularly noticeable for higher dimensions (Dim=20), where standard models (non-FER) require significantly longer execution times, limiting their suitability in educational contexts where speed is crucial, especially for adaptive testing.

TABLE IV. COMPARISON OF THE PERFORMANCE AND EXECUTION TIMES OF THE DIFFERENT MODELS ON THE ROBOMISSION DATASET. METRICS ARE AVERAGED OVER 5 FOLDS (5-FOLD CROSS-VALIDATION). THE ↑ SYMBOL INDICATES THAT A HIGHER VALUE IS BETTER, WHILE ↓ INDICATES THAT A LOWER VALUE IS PREFERABLE. VALUES IN BOLD REPRESENT THE PROPOSED METHOD'S RESULTS, WHEREAS THE OTHERS ARE TAKEN FROM THE STUDY BY BENOÎT CHOFFIN (2021).

Model	Dim	Results AUC ↑	Execution Time (TE) ↓	Results RMSE ↓	Results NLL ↓
DAS3H	0	0,851	2min55s	0,273	0,252
FER-DAS3H	0	0,828	14s	0,286	0,272
DAS3H	5	0,849	21min46s	0,277	0,256
FER-DAS3H	5	0,825	4min26s	0,276	0,254
DAS3H	20	0,853	1h8min25s	0,275	0,253
FER-DAS3H	20	0,826	4min16s	0,287	0,272
PFA	0	0,828	1min7s	0,275	0,26
FER-PFA	0	0,820	4s	0,281	0,268
PFA	5	0,849	2min42s	0,269	0,247
FER-PFA	5	0,821	1min3s	0,281	0,267
PFA	20	0,851	7min36s	0,267	0,245
FER-PFA	20	0,821	2min12s	0,281	0,267
AFM	0	0,793	11s	0,287	0,280
FER-AFM	0	0,815	0s	0,283	0,270
AFM	5	0,815	2min39s	0,282	0,269
FER-AFM	5	0,814	1min1s	0,283	0,270
AFM	20	0,818	6min37s	0,281	0,267
FER-AFM	20	0,814	2min10s	0,283	0,270

TABLE V. COMPARISON OF THE PERFORMANCE AND EXECUTION TIMES OF THE DIFFERENT MODELS ON THE ALGEBRA05 DATASET. METRICS ARE AVERAGED OVER 5 FOLDS (5-FOLD CROSS-VALIDATION). THE ↑ SYMBOL INDICATES THAT A HIGHER VALUE IS BETTER, WHILE ↓ INDICATES THAT A LOWER VALUE IS PREFERABLE. VALUES IN BOLD REPRESENT THE PROPOSED METHOD'S RESULTS, WHEREAS THE OTHERS ARE TAKEN FROM THE STUDY BY BENOÎT CHOFFIN (2021).

Model	Dim	Results AUC ↑	Execution Time (TE) ↓	Results RMSE ↓	Results NLL ↓
DAS3H	0	0,829	8min16s	0,361	0,411
FER-DAS3H	0	0,827	6min9s	0,361	0,412
DAS3H	5	0,828	53min40s	0,362	0,412
FER-DAS3H	5	0,826	48min58s	0,363	0,413
DAS3H	20	0,827	3h55min52s	0,363	0,414
FER-DAS3H	20	0,823	2h24min21s	0,365	0,419
PFA	0	0,754	8min47s	0,390	0,468
FER-PFA	0	0,750	2min47s	0,390	0,472
PFA	5	0,767	38min54s	0,385	0,458
FER-PFA	5	0,746	19min23s	0,392	0,474
PFA	20	0,768	1h12min27s	0,385	0,458
FER-PFA	20	0,746	45min2s	0,392	0,474
AFM	0	0,721	2min38s	0,398	0,486
FER-AFM	0	0,720	1min45s	0,398	0,486
AFM	5	0,723	18min29s	0,395	0,481
FER-AFM	5	0,715	11min57s	0,400	0,490
AFM	20	0,723	49min55s	0,395	0,481
FER-AFM	20	0,720	27min50s	0,400	0,490

For the "Robomission" dataset (Table IV), the FER-MODELS show substantial gains in speed. For example, with an embedding dimension of 20, the DAS3H model requires 1h8min25s, while FER-DAS3H reduces complexity and executes the model in only 4min16s, a dramatic reduction in execution time. Although this optimization comes with a slight decrease in performance (AUC from 0.853 to 0.826), the gap remains modest, making FER-DAS3H an ideal choice when execution speed is critical, particularly for adaptive testing. Similarly, for the PFA models, transitioning to the FER version reduces execution time from 7min 36s to 2min 12s for Dim=20, while maintaining an AUC close to 0.82, demonstrating the efficiency of these optimized versions.

In the "Algebra05" dataset (Table V), the reduction in execution times achieved by the FER models is equally impressive. For Dim=20, the DAS3H model takes nearly 4 hours (3h55min52s) to produce results, whereas FER-DAS3H reduces the complexity and completes the task in 2h24min21s, nearly twice as fast. A similar observation holds for FER-PFA compared to PFA, which completes the analysis in 45min2s compared to 1h12min27s for the standard version. These speed gains make the FER models appealing for educational applications, even though their AUC is slightly lower, as seen in the difference between 0.768 (PFA) and 0.746 (FER-PFA) for Dim=20.

TABLE VI. COMPARISON OF THE PERFORMANCE AND EXECUTION TIMES OF THE DIFFERENT MODELS ON THE BRIDGE_ALGEBRA06 DATASET. METRICS ARE AVERAGED OVER 5 FOLDS (5-FOLD CROSS-VALIDATION). THE ↑ SYMBOL INDICATES THAT A HIGHER VALUE IS BETTER, WHILE ↓ INDICATES THAT A LOWER VALUE IS PREFERABLE. VALUES IN BOLD REPRESENT THE PROPOSED METHOD'S RESULTS, WHEREAS THE OTHERS ARE TAKEN FROM THE STUDY BY BENOÎT CHOFFIN (2021).

Model	Dim	Results AUC ↑	Execution Time (TE) ↓	Results RMSE ↓	Results NLL ↓
DAS3H	0	0,802	30min19s	0,333	0,362
FER-DAS3H	0	0,802	25min13s	0,333	0,362
DAS3H	5	0,794	1h57min8s	0,336	0,367
FER-DAS3H	5	0,794	1h30min7s	0,336	0,369
DAS3H	20	0,794	7h5min58s	0,335	0,367
FER-DAS3H	20	0,794	6h14min24s	0,335	0,367
PFA	0	0,748	9min52s	0,348	0,394
FER-PFA	0	0,751	7min9s	0,346	0,392
PFA	5	0,756	1h1min50s	0,346	0,390
FER-PFA	5	0,752	53min14s	0,347	0,393
PFA	20	0,756	2h50min4s	0,346	0,390
FER-PFA	20	0,752	1h52min2s	0,347	0,393
AFM	0	0,709	10min10s	0,356	0,410
FER-AFM	0	0,709	9min59s	0,356	0,410
AFM	5	0,709	32min5s	0,356	0,410
FER-AFM	5	0,709	31min0s	0,356	0,411
AFM	20	0,709	2h3min51s	0,356	0,410
FER-AFM	20	0,709	1h4min5s	0,356	0,410

TABLE VII. COMPARISON OF THE PERFORMANCE AND EXECUTION TIMES OF THE DIFFERENT MODELS ON THE ASSISTMENTS17 DATASET. METRICS ARE AVERAGED OVER 5 FOLDS (5-FOLD CROSS-VALIDATION). THE ↑ SYMBOL INDICATES THAT A HIGHER VALUE IS BETTER, WHILE ↓ INDICATES THAT A LOWER VALUE IS PREFERABLE. VALUES IN BOLD REPRESENT THE PROPOSED METHOD'S RESULTS, WHEREAS THE OTHERS ARE TAKEN FROM THE STUDY BY BENOÎT CHOFFIN (2021).

Model	Dim	Results AUC ↑	Execution Time (TE) ↓	Results RMSE ↓	Results NLL ↓
DAS3H	0	0,682	35min59s	0,460	0,611
FER-DAS3H	0	0,683	9min37s	0,460	0,611
DAS3H	5	0,682	3h22min22s	0,465	0,621
FER-DAS3H	5	0,677	55min54s	0,464	0,620
DAS3H	20	0,675	4h22min46s	0,469	0,629
FER-DAS3H	20	0,666	1h58min25s	0,477	0,646
PFA	0	0,621	6min13s	0,474	0,64
FER-PFA	0	0,633	4min50s	0,471	0,634
PFA	5	0,641	47min11s	0,469	0,631
FER-PFA	5	0,635	22min12s	0,47	0,633
PFA	20	0,643	1h32min26s	0,469	0,63
FER-PFA	20	0,635	51min37s	0,47	0,633
AFM	0	0,591	4min30s	0,478	0,649
FER-AFM	0	0,616	4min28s	0,474	0,640
AFM	5	0,616	27min9s	0,474	0,640
FER-AFM	5	0,616	22min59s	0,474	0,640
AFM	20	0,618	1h0min39s	0,473	0,639
FER-AFM	20	0,616	49min22s	0,474	0,640

For the "Bridge_algebra06" dataset (Table VI), the time difference between the standard and FER versions is also noteworthy. The DAS3H model with an embedding dimension of 5 takes more than two hours (1h57min8s) to process the data, whereas FER-DAS3H reduces complexity and completes the task in 1h30min7s, a 25% reduction. For an embedding dimension of 20, FER-DAS3H continues to significantly reduce time, from nearly 7 hours (7h5min58s) to 6h14min24s, saving almost an hour. These results confirm that FER versions not only accelerate computations but also maintain stable AUC values around 0.794, comparable to non-FER versions.

Finally, on the "Assistments17" dataset (Table VII), the Fast E-learning Recommendation models once again demonstrate their superiority in terms of speed. For an embedding dimension of 20, the DAS3H model requires more than 4 hours (4h22min46s) to generate results, while FER-DAS3H reduces complexity and completes the task in less than 2 hours (1h58min25s), cutting the time in half. A similar trend is observed for the AFM models, where FER-AFM completes calculations in 49min22s compared to 1h0min39s for the standard version. This acceleration is especially valuable in contexts requiring rapid iterations or real-time interactions.

Overall, the FER versions demonstrate remarkable capability in reducing execution times, making these models particularly suitable for situations where time constraints outweigh marginal performance gains. While standard versions achieve slightly higher AUC, the speed improvements offered by the FER versions enable broader and more flexible use, especially for large-scale datasets or computationally intensive tasks. This efficiency positions the proposed FER versions as compelling alternatives that balance performance and speed.

The impact of embedding dimension d on predictive performance has often been addressed in terms of model accuracy, as evidenced by a 2021 study focused on AUC evaluation (Area Under the Curve). However, the effect of this dimension on model training time remains underexplored, despite its critical importance, particularly in contexts with limited computational resources.

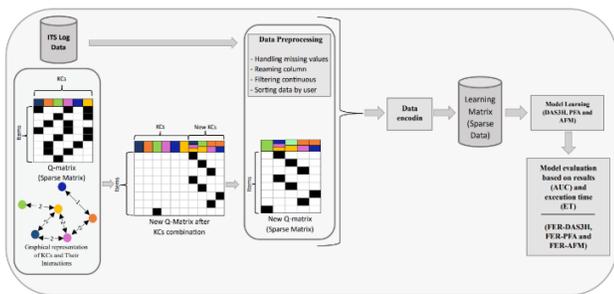


Fig. 1. FER-MODEL Learner performance prediction framework.

This issue is analyzed in the present study by examining how increasing the embedding dimension d affects model training speed. The results show that increasing d significantly raises the time required to train standard models. In contrast, the proposed approach demonstrates a remarkable ability to reduce this time, even with higher embedding dimensions. This improvement is particularly evident as d increases, where the gap between the training time of the proposed method and that of standard

models becomes more pronounced. These results highlight not only the efficiency of the proposed approach in terms of speed but also its potential for handling datasets that require high-dimensional vector representations without compromising predictive performance (see Fig. 1).

VI. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

This study highlighted the benefits of a structured simplification of the Q-matrix in knowledge tracing models, aiming to reduce execution time without compromising predictive quality. The proposed approach, named Fast E-learning Recommendation (FER), is based on merging all knowledge components associated with each item into a single combined skill. This transformation reduces the density of the Q-matrix and significantly accelerates model training. The results obtained from multiple real-world datasets demonstrate that this simplification preserves and in some cases enhances predictive performance while reducing computational costs.

However, the proposed approach also presents certain limitations. Notably, the current strategy involves systematically combining all knowledge components associated with an item, which may lead to a loss of granularity in skill representation. While this global simplification yields considerable gains in computation time, it does not guarantee optimal preservation of pedagogical information. In some cases, merging all components might obscure subtle but educationally meaningful distinctions between learners' skill sets.

To address this limitation, a promising direction for future research would be to develop a more selective and controlled simplification strategy for the Q-matrix. Rather than merging all associated components, only a subset could be combined, guided by criteria such as redundancy, correlation between skills, or their impact on predictive accuracy. This would enable an optimal trade-off between the expressiveness of the Q-matrix and computational efficiency.

Future work could also explore adaptive simplification mechanisms capable of dynamically adjusting the Q-matrix structure based on the characteristics of the dataset. Additionally, integrating this transformation with feature selection techniques or multi-task learning frameworks could enhance model robustness, especially in heterogeneous educational environments.

STATEMENTS ON OPEN DATA AND ETHICS

The ASSISTments and KDD datasets used in this study are publicly available on the website <https://sites.google.com/site/assistmentsdata>, and the Robomission19 database is available on GitHub with the reference: (Adaptive Learning Group). Additional information about this dataset can be found in the associated article (Benoit Choffin, 2021).

REFERENCES

- [1] John R. Anderson, C. Franklin Boyle and BJR 1985 Intelligent Tutoring System 456–462
- [2] Nwana H S 1990 Intelligent Tutoring Systems: an overview vol 4
- [3] Taylor, D.L., Yeung, M., Basset A Z 2021 Personalized and Adaptive Learning, Innovative Learning Environments in STEM Higher Education pp 17–34, Springer Briefs in Statistics

- [4] Becker S et al 2018 Horizon Report 2018 Higher Education Edition 1–60
- [5] Wang T H 2007 What strategies are effective for formative 171–186
- [6] Canfield W 2001 ALEKS: A Web-based intelligent tutoring system
- [7] Republic C 2019 Blockly Programming Dataset 3–5
- [8] Sacksick M 2019 Evaluer les apprentissages : comparaison de deux techniques de diagnostic de connaissance Eiah 73–8
- [9] Pavlik P I, Cen H and Koedinger K R 2009 Performance factors analysis - A new alternative to knowledge tracing *Frontiers in Artificial Intelligence and Applications* vol 200 (IOS Press) pp 531–8
- [10] Cen H, Koedinger K and Junker B 2006 Learning factors analysis - A general method for cognitive model evaluation and improvement *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* 4053 LNCS 164–75
- [11] Cen H, Koedinger K and Junker B 2008 Comparing Two IRT models for conjunctive skills *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* 5091 LNCS 796–8
- [12] Choffin B, Popineau F, Bourda Y and Vie J J 2019 DAS3H: Modeling student learning and forgetting for optimally scheduling distributed practice of skills *EDM 2019 - Proceedings of the 12th International Conference on Educational Data Mining* pp 29–38
- [13] Wilson M and De Boeck P 2004 Descriptive and explanatory item response models *Explanatory Item Response Models* pp 43–74
- [14] Corbett A T and Anderson J R 1995 Knowledge Tracing User Model. *User-adapt. Interact.* 4 253–78
- [15] TATSUOKA K K 1983 Rule Space: an Approach for Dealing With Misconceptions Based on Item Response Theory *J. Educ. Meas.* 20 345–54
- [16] Zhang M, Zhu X, Zhang C, Pan F, Qian W and Zhao H 2023 No Length Left Behind: Enhancing Knowledge Tracing for Modeling Sequences of Excessive or Insufficient Lengths *International Conference on Information and Knowledge Management, Proceedings* pp 3226–35
- [17] Hakkal S and Ayoub Ait Lahcen 2024 XGBoost To Enhance Learner Performance Prediction *Comput. Educ. Artif. Intell.* 7
- [18] Chen T and Guestrin C 2016 XGBoost: A scalable tree boosting system *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 13-17-Aug 785–94
- [19] Wang S, Liu J and Shroff N 2018 Fast sparse matrix multiplication *35th Int. Conf. Mach. Learn. ICML 2018* 12 8176–93
- [20] Schmucker R, Wang J, Hu S and Mitchell T M 2022 Assessing the Performance of Online Students - New Data, New Approaches, Improved Accuracy *J. Educ. Data Min.* 14 1–45
- [21] Wim J. van der Linden 2014 *Handbook of Item Response Theory Modeling Handb. Item Response Theory Model.*
- [22] Lindsey R V., Shroyer J D, Pashler H and Mozer M C 2014 Improving Students' Long-Term Knowledge Retention Through Personalized Review *Psychol. Sci.* 25 639–47
- [23] Rendle S 2012 Factorization machines with libFM *ACM Trans. Intell. Syst. Technol.* 3
- [24] Benoi and Choffin 2021 Algorithmes d'espacement adaptatif de l'apprentissage pour l'optimisation de la maîtrise à long terme de composantes de connaissance
- [25] De la Torre J 2009 DINA model and parameter estimation: A didactic *J. Educ. Behav. Stat.* 34 115–30
- [26] Barnes T 2005 The Q-matrix method: Mining student response data for knowledge *AAAI Work. - Tech. Rep. WS-05-02* 39–46
- [27] Weinberger K, Attenberg J and Org A S 2009 Feature Hashing for Large Scale Multitask Learning
- [28] Koedinger K R, Corbett A T and Perfetti C 2012 The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning *Cogn. Sci.* 36 757–98
- [29] Legendre P 2007 Principal Component Analysis (PCA) *Principal Component Analysis (PCA) Statistics (Ber.)* 1–12
- [30] Hayakawa M 2015 Matrix Factorization Techniques for Recommender Systems *Earthq. Predict. with Radio Tech.* 199–207
- [31] Feng M, Heffernan N and Koedinger K 2009 Addressing the assessment challenge with an online system that tutors as it assesses *User Model. User-adapt. Interact.* 19 243–66
- [32] Gervet T, Koedinger K, Schneider J and Mitchell T 2020 When is deep learning the best approach to knowledge tracing? *J. Educ. Data Min.* 12 31–54
- [33] Stamper J and Pardos Z A 2010 The 2010 KDD Cup Competition Dataset: Engaging the machine learning community in predictive learning analytics *J. Learn. Anal.* 3 312–6
- [34] Lai Z, Wang L and Ling Q 2021 Recurrent knowledge tracing machine based on the knowledge state of students *Expert Syst.* 38 1–18
- [35] Choffin B and Ueda N 2018 Scaling Bayesian Optimization up to Higher Dimensions: A Review 2018 *IEEE 28th Int. Work. Mach. Learn. Signal Process.* 1–6
- [36] Defazio A, Bach F and Lacoste-Julien S 2014 SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives *Advances in Neural Information Processing Systems* vol 2 pp 1646–54