Modelling Learners in Adaptive Educational Systems: A Multivariate Glicko-based Approach

SOLMAZ ABDI, HASSAN KHOSRAVI, and SHAZIA SADIQ, The University of Queensland, Australia

The Elo rating system has been recognised as an effective method for modelling students and items within adaptive educational systems. A common characteristic across Elo-based learner models is that they are not sensitive to the lag time between two consecutive interactions of a student within the system. Implicitly, this characteristic assumes that students do not learn or forget between two consecutive interactions. However, this assumption seems insufficient in the context of adaptive learning systems where students could have improved their mastery through practising outside of the system or that their mastery may be declined due to forgetting. In this paper, we extend the existing works on the use of rating systems for modelling learners in adaptive educational systems by proposing a new learner model called MV-Glicko that builds on the Glicko rating system. MV-Glicko is sensitive to the lag time between two consecutive interactions of a student within the system and models it as a parameter that captures the confidence of the system in the current inferred rating. We apply MV-Glicko on three public data sets and three data sets obtained from an adaptive learning system and provide evidence that MV-Glicko outperforms other conventional models in estimating students' knowledge mastery.

CCS Concepts: • **Applied computing** → **Education**; **E-learning**; • **Information systems** → **Data mining**; • **Human-centered computing** → **User models**;

Additional Key Words and Phrases: Adaptive learning, learner modelling, knowledge tracing, Glicko rating system

ACM Reference Format:

Solmaz Abdi, Hassan Khosravi, and Shazia Sadiq. 2021. Modelling Learners in Adaptive Educational Systems: A Multivariate Glicko-based Approach. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3375462.3375520

1 INTRODUCTION

Adaptive educational systems (AES) make use of data about students, learning process, and learning products to adapt the level or type of instruction for each student. Commonly, this adaptivity takes the form of selecting items from a large repository of learning items to match the current learning ability of a student [18]. To do so, adaptive educational systems rely heavily on a component called learner model that captures an abstract representation of a student's ability level based on their performance and interactions with the educational system [8].

The use of a Rating System (RS) and in particular Elo rating system has been widely studied for modelling students' learning in AESs [2, 15, 20, 21]. In the educational setting, the Elo rating system is used as a learner model to conduct a paired comparison between students and learning items as two opponents competing with each other [21, 28]. The advantage of using Elo-based learner models is that despite being heuristic, they are simple, fast, order-sensitive, and robust in modelling students and items [20]. These benefits make Elo-based models desirable for the development of adaptive educational systems where it is required to update students' proficiency level upon administration of each learning item in real time [28]. In addition, Elo-based learner models provide explicit and interpretable estimations

Manuscript submitted to ACM

about students' knowledge states and difficulty of learning items [3], which makes them easy to be opened to students and other stakeholders based on the principles of open learner models (OLMs) [4].

Elo-based models, as well as many of the conventional learner models (e.g., Bayesian Knowledge Tracing (BKT) [7], Item Response Theory (IRT) [26], Additive Factor Model (AFM) [5], and Performance Factor Analysis (PFA) [19]), are not sensitive to the lag time between two consecutive interactions of a student within the system. Implicitly, this characteristic assumes that students do not learn or forget knowledge between two consecutive interactions. This assumption may be reasonable in the context of adaptive testing, where a student takes the test in one sitting within a short time frame without receiving feedback on their answers. However, it seems less reasonable in the context of adaptive learning systems where a student could have enhanced their knowledge proficiency by engaging with learning materials and exercises outside of the system or that their knowledge proficiency may have declined due forgetting [10]. As a result, many of the conventional learner models are extended to become sensitive to lag time between two consecutive interactions of a student within the system (e.g., [6, 17, 23]).

The goal of the present article is to overcome the presented limitations of the existing Elo-based learner models by proposing a new learner model called MV-Glicko that is built based on the principles of Glicko rating system [12]. Similar to Elo-based learner models, MV-Glicko estimates the students' knowledge states and item difficulties by conducting a paired comparison between students and items. However, the distinction between MV-Glicko and its Elo-based counterparts is that MV-Glicko does not assume that the estimation about a student's knowledge state is deterministic with regards to time if the student does not engage with the system. Instead, when updating a student's knowledge state, MV-Glicko takes into account the lag time between consecutive interactions of the student within the system and models this lag time as a parameter that captures the confidence of the model about its current inferred estimations. Formally, instead of using a single number for reporting the ratings, as is the case in Elo-based learner models, MV-Glicko reports a student's rating (knowledge state) by a 95% confidence interval. A longer lag time between two consecutive interactions of a student leads to an increase in the range of the confidence interval, suggesting that the model has become less confident about its inferred rating for the student. To evaluate the model, we examine the predictive performance of MV-Glicko on three public data sets. Results suggest that MV-Glicko outperforms other conventional learner models in predicting students' performance. To evaluate the MV-Glicko in the context of adaptive learning, we examine its predictive performance on three data sets obtained from an authentic online adaptive learning platform. Results demonstrate that MV-Glicko provides superior predictive performance compared to conventional models and is suitable to be embedded in online adaptive educational systems. Finally, we conduct a sensitivity analysis to investigate the impact of different parameters of the model on its overall performance.

2 RELATED WORK

Conventional approaches. Two conventional approaches have been widely studied for the development of learner models. The first common approach is Knowledge Tracing (KT) that uses the sequences of students' interaction with the system to model the evolution of their knowledge state over time and predict their future sequence of responses [6]. The leading model in this category is Bayesian Knowledge Tracing (BKT) [7] that uses a Hidden Markov Model (HMM) to capture students' knowledge state through a set of binary variables that indicate whether a knowledge state has been mastered by a student or not. One important way in which BKT has been extended more recently is to replace the Hidden Markov model with a recurrent neural network (RNNs), often referred to as deep knowledge tracing (DKT) [22]. DKT that captures complex representations of students' knowledge state using long short-term memory model (LMST) showed promising results in predicting students' performance compared to BKT [22]. The second common approach

for learner modelling is Factor Analysis that in contrast to KT approaches, does not take the sequence of observations into account; rather, it uses the collection of observations to learn a set of generalisable factors about data [27]. The leading learner models in this category includes Item Response Theory (IRT) [26] and its extensions Additive Factor Model (AFM) [5], and Performance Factor Analysis (PFA) [19]. Neither of KT or Factor Analysis approaches, however, can be easily implemented into online adaptive educational systems as they generally require pre-calibration on big samples of data and ongoing addition of new students and new learning items to the system necessitates nontrivial continuous calibration of model parameters [20].

Rating Systems. Using a rating system and in particular, the Elo rating system is known as an effective alternative learner modelling approach to the aforementioned conventional models. The Elo rating system is originally developed for rating chess players and is developed based on the paired comparison of data about two chess players when they compete against each other [28]. As a learner model in educational settings, the Elo rating system conducts a similar paired comparison between students and learning items. This means that when a student attempts an item, the model considers them as two rivals competing with each other. The initial implementation of Elo as a learner model known as standard Elo-based model is similar to IRT, where a student's mastery and the item difficulty are modelled using two global parameters [21]. Two important extensions over the standard Elo-based model are the multivariate Elo-based model [9] and hierarchical model [21], where instead of using a global knowledge parameter for students, they use an overlay model which estimates the competency of a student in each different concept using a separate parameter. More recently, [2] proposed an extension over these two models called M-Elo that models students and items in the presence of items with multiple concepts. A common feature across all Elo-based learner models is that they are not sensitive to the lag time between the interactions of a student. So, they implicitly assume that no learning or forgetting happens for the student during that time interval. Glicko rating system [11] is another variant of rating systems originally proposed to overcome the limitations of the Elo rating system in the context of chess games and other tournaments. In particular, unlike Elo rating system that computes a single rating for each player, Glicko rating system computes both a rating indicating the mastery of the player and a rating deviation (RD) representing the confidence of the model about the estimated rating for the player[12]. Glicko then uses the rating and the RD to report a 95% confidence interval about the player's mastery. A high value of RD means less confidence in the inferred rating and suggests that either the player has participated in a small number of games or that the player does not compete frequently and the lag time between consecutive games of the player is high [11]. Despite its potential advantages, the use of Glicko-based learner model has received little attention. Recently, [24] proposed using a Glicko-based learner model for modelling the mastery of students studying within the Coursera platform. However, similar to the standard Elo-based learner model, their proposed model considers only one global parameter for estimating a student's knowledge state on the entire domain and that their model is also not sensitive to the lag time between a student's interactions.

3 GLICKO-BASED LEARNER MODELLING

In this section, we present our proposed Glicko-based learner model, MV-Glicko. In what follows, Section 3.1 introduces mathematical notation and a formal definition of the problem under investigation, and Section 3.2 introduces MV-Glicko.

3.1 Notations, Assumptions, and Problem formulation

Notations Let $S_N = \{s_1 \dots s_N\}$ represents a set of students who are enrolled in a course in an AES, where s_n refers to an arbitrary student. Each course offered by the AES covers a set of concepts $\Delta_C = \{\delta_1 \dots \delta_C\}$, referred to as the domain model, where δ_c presents an arbitrary concept. In this work, the notion of a concept is based on taxonomies of

knowledge components described by [16]. Furthermore, let $Q_M = \{q_1 \dots q_M\}$ denote the content model, representing a repository of learning resources that are available to students within the AES, where q_m refers to an arbitrary resource; Let $\Omega_{M\times C}$ denote the existing association between each learning resource and concepts of the course, where ω_{mc} is $1/f$ if item q_m is tagged with f concepts including δ_c, and 0 otherwise. Let t index the timestamp of a student interaction with the AES. Since we only require the time difference between two interactions, timestamps are presumed to be recorded as the fraction of days passed since the first interaction of the student with the course within the AES. Finally, let's assume that the AES records the interaction log for s_n as $i_t = (s_n, q_m, t, a_{nm})$, where a_{nm} indicates the correctness of the student s_n attempt on item q_m at timestamp $t \in \mathbb{R}^+$ ($a_{nm} = 1$ indicates that at timestamp t student s_n has answered item q_m correctly, and $a_{nm} = 0$ indicates that student s_n has answered item q_m incorrectly). As such, s_n's interaction log can be modelled as a sequence $L = \{i_1, i_2, ..., i_t\}$.

Problem Statement. Given a sequence of students' interaction L, our aim is to infer (1) a learner model that estimates each student s_n's knowledge state on each concept δ_c and (2) the difficulty level of each item q_m.

3.2 Multivariate Glicko-based Learner Model (MV-Glicko)

MV-Glicko is derived from the Glicko rating system [12] and extend it to be used in the educational settings. In particular, the Glicko rating system considers one parameter to model the mastery of each player. In the educational setting, this implies that having a homogeneous domain where, for each student, one global parameter indicates the proficiency level of the student on the entire domain. However, in practice, the domain of an AES is generally made-up of different knowledge components, and each item that exists in the repository of the system might be associated with one or multiple of those knowledge components. Accordingly, MV-Glicko modifies the formulation of the Glicko rating system to fit it for the context of AESs, where there exist multiple concepts in the domain, and each learning item might be associated with multiple knowledge components. To do so, MV-Glicko uses an overlay model which estimates the competency of a student in each different concept using a separate parameter and a global parameter for modelling the difficulty of each item. For each student s_n on each concept δ_l, the estimation of MV-Glicko about the student's knowledge state is represented as a 95% confidence interval using two parameters: (1) λ_{nl}, which represents s_n's mean rating on δ_n, and (2) σ_{nl}, refereed to as the "rating deviation", which represents the standard deviation that captures the student's estimated knowledge state by 95% confidence interval, where the lowest value in the interval indicates the student's rating minus twice the rating deviation, and the highest value is the student's rating plus twice the rating deviation. For example, if $\lambda_{nl} = 1600$ and $\sigma_{nl} = 30$, we would say that we're 95% confident that the actual s_n's rating on δ_{nl} lies between 1540 and 1660. Similarly, the difficulty of each arbitrary learning item q_m computed by MV-Glicko is composed of two parts: d_m that represents the rating of q_m which should be interpreted as the item difficulty, and (2) σ_m which represents the "rating deviation" of MV-Glicko about the difficulty of q_m. A higher value of σ_m indicates that q_m has been attempted only by a few students, and MV-Glicko is still not confident about the actual difficulty of the learning item. Whenever s_n attempts q_m, MV-Glicko takes a four-step approach to update s_n's knowledge state on each concept associated with q_m and updating the difficulty of q_m.

Step 1: Determining rating deviation for the student. For each concept δ_n that q_m is tagged with, MV-Glicko first examines its previous estimations about the knowledge state of s_n on δ_n. If it is the first time that s_n encounters concept δ_l, MV-Glicko initialises the values of λ_{nl} and σ_{nl}. MV-Glicko assumes that before attempting any items associated with δ_l by s_n, their knowledge state on δ_l follows a normal distribution with mean λ_0 and variance σ_0^2 ($N(\lambda_0, \sigma_0^2)$). Ideally, the value of λ_0 and σ_0^2 should be inferred from data by optimising the predictability of the outcome of students' attempts on items. However, with the absence of any initial information about students in AESs, the initial rating

and rating deviation for all students on all concepts are set to 1500 and 350, respectively, as the reasonable default choices recommended by [11]. Otherwise, if s_n has encountered δ_n previously, MV-Glicko updates its belief about the knowledge state of s_n on δ_l by measuring the lag time between the current interaction and the s_n's previous interaction that was associated with δ_l. MV-Glicko assumes that if time passes and s_n does not practice any items associated with concept δ_l, its existing estimation about s_n's knowledge state on δ_l is less reliable. This, in turn, is reflected in the estimated rating deviation for the student on that concept (σ_{nl}) increasing. Assuming that at the previous interaction which we denote with timestamp t_0, s_n's knowledge state is distributed as $N(\lambda_{nl}, \sigma^2_{nl})$, with the passage of t units of time without any attempts from s_n on δ_l, the student's knowledge state distribution on δ_n is updated as $N(\lambda_{nl}, \sigma^2_{nl} + v^2 t)$, where v^2 is the increase in variance per unit of time and needs to be inferred from the data. As recommended by [11], $\sigma_{nl} = min(\sqrt{(\sigma^2_{nl_{(t_0)}} + v^2 t)}, 350)$ is used by MV-Glicko to update the value of rating deviation with the passage of time. The reason for setting 350 as the maximum value of rating deviation is to ensure that the updated rating deviation is never bigger than the initial rating deviation for students when they had no attempts on the system.

Step 2: Determining rating deviation for the item: Similar to Step 1, for learning item q_m, MV-Glicko examines if q_m has been previously encountered by students. If it is the first time q_m is being attempted by a student, MV-Glicko initialises the values of d_m and σ_m. MV-Glicko assumes that the initial difficulty of each learning item follows a normal distribution with mean d_0 and variance σ_0^2 ($N(d_0, \sigma_0^2)$). Ideally, the value of d_0 and σ_0^2 should be inferred from data, but similar to the explanations provided in Step 1, the values of d_0 and σ_0^2 are initialised to 1500 and 350 for all items as reasonable default choices. Otherwise, MV-Glicko restores its previous estimations about d_m and σ_m to be used in Step 3 and Step 4. MV-Glicko does not update the rating deviation in the estimated difficulty of items with the passage of time, as the estimations of MV-Glicko about the difficulty of the learning items are not expected to become less reliable when students do not attempt the item for a while.

Step 3: Updating the student's knowledge states. After MV-Glicko restores and rectifies its prior estimations about s_n and q_m as it was explained in Step 1 and Step 2, it uses the outcome of s_n's attempt on q_m to update its estimations about the student's ratings and the difficulty of q_m. In what follows, a_{nm} indicates the outcome of s_n's attempt on q_m which equals to 1 if the student answers q_m correctly, and 0 otherwise. To update its estimations about s_n's rating on each concept δ_l associated with q_m, MV-Glicko uses the following formulation:

$$\lambda_{nl} := \begin{cases} \lambda_{nl} + K \cdot (1 - E(a_{nm} = 1 | \lambda_{nl}, d_m, \sigma_m)), & \text{if the answer is correct } (a_{nm} = 1) \\ \lambda_{nl} + K \cdot \eta \cdot (-E(a_{nm} = 1 | \lambda_{nl}, d_m, \sigma_m)), & \text{if the answer was incorrect } (a_{nm} = 0) \end{cases} \quad , \quad \sigma_{nl} := \sqrt{(\frac{1}{\sigma^2_{nl}} + \frac{1}{\gamma^2_{nl}})^{-1}} \tag{1}$$

Where η is a constant value to control the sensitivity of estimations based on the latest attempt if the answer was incorrect, and

$$q = \frac{\ln 10}{400}, \qquad g(\sigma_m) = \frac{1}{\sqrt{1 + 3q^2\sigma_m^2/\pi^2}}, \qquad E(a_{nm} = 1 | \lambda_{nl}, d_m, \sigma_m) = \frac{1}{1 + 10^{-g(\sigma_m)(\lambda_{nl} - d_m)/400}}$$

$$\gamma^2_{nl} = (q^2 \cdot (g(\sigma_m))^2 \cdot E(a_{nm} = 1 | \lambda_{nl}, d_m, \sigma_m) \cdot (1 - E(a_{nm} = 1 | \lambda_{nl}, d_m, \sigma_m)))^{-1}, \qquad K = max(\frac{q}{1/\sigma^2_{nl} + 1/\gamma^2_{nl}} \cdot g(\sigma_m), K')$$

Where K' is a constant determining a lower boundary for the amount of updates to the student's rating. The reason for considering a lower boundary is that when a student frequently interacts with the AES, MV-Glicko becomes more certain about the student's knowledge state, so, the changes to the student's rating becomes very slow. This has shown to discourage students' engagement with the AES, particularly, if the learner model is opened to students based on the principals of open learner models [2]. By setting a minimum threshold for K, we ensure that the student's rating is updated appreciably, even if the student interacts very frequently with the AES. In addition, $E(a_{nm} = 1 | \lambda_{nl}, d_m, \sigma_m)$ represents the expected outcome of the student's attempt on q_m from the student's perspective.

Step 4: Updating the difficulty of the learning item. To update the difficulty of q_m, the model first computes $\bar{\lambda}_{nm} = \sum_{l=1}^{L} \lambda_{nl} \times \omega_{ml}$ and $\bar{\sigma}_{nm} = \sum_{l=1}^{L} \sigma_{nl} \times \omega_{ml}$ to estimate s_n's average competency on the concepts associated with q_m, and the average of their corresponding rating deviations, respectively. It then updates the difficulty of q_m and its corresponding rating deviation using the following formulas:

$$d_m := d_m + \frac{q}{1/\sigma_m^2 + 1/\gamma_m^2} \cdot g(\bar{\sigma}_{nm}) \cdot (a^o_{nm} - E(a^o_{nm} = 1|\bar{\lambda}_{nm}, d_m, \bar{\sigma}_{nm})), \quad \sigma_m := \sqrt{(\frac{1}{\sigma_m^2} + \frac{1}{\gamma_m^2})^{-1}} \tag{2}$$

Where a^o_{nm} is the result of the attempt from q_m perspective, i.e., it is 0 if the student answers the item correctly and is 1 if the student answers it incorrectly, and

$$g(\bar{\sigma}_{nm}) = \frac{1}{\sqrt{1 + 3q^2\bar{\sigma}_{nm}^2/\pi^2}}, \qquad E(a^o_{nm} = 1|d_m, \bar{\lambda}_{nm}, \bar{\sigma}_{nm}) = \frac{1}{1 + 10^{-g(\bar{\sigma}_{nm})(d_m - \bar{\lambda}_{nm})/400}}$$
$$\gamma_m^2 = (q^2 \cdot (g(\bar{\sigma}_{nm}))^2 \cdot E(a^o_{nm} = 1|d_m, \bar{\lambda}_{nm}, \bar{\sigma}_{nm}) \cdot (1 - E(a^o_{nm} = 1|d_m, \bar{\lambda}_{nm}, \bar{\sigma}_{nm})))^{-1}$$

It should be noted that the updates to the students' knowledge states (Equation 1) and the difficulty of items (Equation 2) are performed simultaneously. Please refer to [12] for more information about the derivation of the closed form computations used in Step 3 and Step 4 for updating the knowledge state of students and difficulty of items.

4 EVALUATION

We evaluate MV-Glicko using two sets of experiments. In the first experiment, we evaluate the predictive performance of MV-Glicko by comparing it with five conventional learner models on three public educational data sets, and three data sets obtained from an authentic AES called RiPPLE. We then conduct a set of experiments to investigate the sensitivity of MV-Glicko to the different values of the model hyper-parameters. In what follows, Section 4.1 introduces the public data sets and RiPPLE data sets, Section 4.2 explains the experimental setup, Section 4.3 presents the results of the conducted predictive performance analysis, and finally, Section 4.4 presents the results of the conducted sensitivity analysis. The code for MV-Glicko is implemented in Python and is available on GitHub [Blinded Citation] for replicating the results.

4.1 Data Sets

4.1.1 Public Data Sets. We use three publicly available educational data sets: 'Algebra I 2005-2006' (Alg2005), 'Algebra I 2006-2007' (Alg2006) and 'Bridge to Algebra 2006-2007' (BAlg2006). These data sets are obtained from Carnegie Learning's Cognitive Tutor and were made available as "Development" sets in KDD Cup 2010 [25]. As it is recommended by the organisers of the KDD Cup 2010 challenge, in our experiments, we use the concatenation of the problem and the "Step ID" for identifying each learning item. We use the train/test split provided by the challenge organisers, discard interactions related to items that are not explicitly associated with any knowledge components, and discard students whose interactions with the system is less than 5. Finally, we use the combination of user, item, and timestamp to remove the duplicated rows from the data sets. More information about the data sets is provided in Table 1.

4.1.2 RiPPLE Data Sets. We use three data sets obtained from an AES called RiPPLE which recommends learning items to students based on their estimated knowledge states from a pool of learning items [13, 14]. Each learning item in the repository is associated with one or more concepts (KC) covered in the course. The three data sets used for the experiments are (1) Introduction to Information Systems (InfoSys), (2) The Brain and Behavioural Sciences (Neuro), and (3) Biological Fate of Drugs (Pharm). Each of these three data sets is obtained from using RiPPLE in the course for the duration of 13 weeks of the semester. For the experiments, we discarded students whose interaction with RiPPLE was less than 5. These data sets do not incorporate any duplicated rows. To split the data into train set and test set, we

Table 1. Data sets

Type	Data Set	Students	KC	Items	multi-KC[1]	Interactions
Public	Alg2005	572	112	173,650	59,083	609,971
	Alg2006	1,649	713	554,039	31,891	1,824,580
	BAlg2006	1,141	493	129,553	1,212	1,822,680
RiPPLE	InfoSys	422	7	2008	313	47,266
	Neuro	530	21	4,807	379	59,950
	Pharm	111	14	641	24	18,158

sorted the data chronologically, and for each student, their first 80% of interactions on a concept were used as the train set, and the remaining 20% were used as the test set. Overall information about these data sets are provided in Table 1.

4.2 Experimental Setting

Models for comparison. We compare the predictive performance of MV-Glicko to IRT, PFA, and AFM that are explained previously. For this comparison, we used the KTM framework [27] for the implementation of IRT, PFA, and AFM. We also compare the predictive performance of MV-Glicko to two Elo-based learner models: standard Elo and M-Elo [2]. As commonly used in evaluating the predictive performance of learner models, in these experiments, we report the area under the curve (AUC), root mean squared error (RMSE) and accuracy (ACC) for each of the learner models.

Implementation considerations. An important consideration about MV-Glicko is that it calibrates the difficulty of items on the fly. This means that the difficulty of one item for a student that attempts it early in the semester would be different from another student that solves this item later in the semester when the model has a well-established estimate of the item difficulty [24]. In the long run, this may cause differences in the calculations of students' knowledge states who have performed exactly the same within the system. Accordingly, as recommended by [24], to provide reliable estimates of students' knowledge states, we run MV-Glicko twice. First, we run MV-Glicko to get a well-established estimate of difficulties. In the second run, the item difficulties are held fixed, students' knowledge states are initialised, and MV-Glicko is executed again to update students' knowledge states. The reported results for MV-Glicko are based on the second run.We follow the same approach for the other rating system based models, i.e., Elo and M-Elo.

Hyper-parameters values. A grid search was conducted to specify the hyper-parameter ν that determines the increase of rating deviations in students' rating as the result of the passage of time and η that controls the impact of an incorrect answer on the updates to the students' rating for MV-Glicko. Across all experiments on public data sets, the value of ν was set to 50, and η was set to 0.7. For RiPPLE data sets, the value of ν was set to 20, and the value of η was set to 0.7. For Elo and M-Elo, the value of hyper-parameter γ was set to 1.8, and the value of hyper-parameter β was set to 0.05. These hyper-parameter values comes from [2], but as reported by [2] and [21], Elo and M-Elo are not sensitive to the changes in these parameters. The other learner models did not have any specific hyper-parameters that requires tuning.

4.3 Predictive Performance Analysis

4.3.1 Predictive Performance on Public Data Sets. Table 2 compares the predictive performance of each of the models mentioned above. As it is presented, on all three data sets, MV-Glicko outperforms other conventional models based on the three evaluation criteria. The predictive performance of MV-Glicko is followed by M-Elo that is ranked as the second-best performing model. In comparison to M-Elo, MV-Glicko provides +0.0380 AUC improvement on Alg2005, +0.0909 AUC improvement on Alg2006, and +0.0410 AUC improvement on BAlg2006. The ranking of the best performing models suggests that, despite the simplicity and ease of implementation, the models developed using rating system approach are robust in predictive performance and can be considered as practical models for the implementation of real-world AESs. Among the other models, IRT and Elo provide almost similar performance on the three data sets and

[1]multi-KC in Table 1 indicates the number of items tagged with two or more knowledge components (KCs)

Table 2. AUC, ACC, and RMSE for public data sets. ↑ (↓) shows the higher (lower) is the better.

Model	Alg2005			Alg2006			BAlg2006		
	AUC ↑	ACC ↑	RMSE ↓	AUC ↑	ACC ↑	RMSE↓	AUC ↑	ACC ↑	RMSE ↓
IRT	0.7736	0.8154	0.3779	0.6649	0.6171	0.4865	0.7322	0.8236	0.3638
AFM	0.6924	0.7912	0.4180	0.5761	0.6047	0.4519	0.6567	0.8249	0.3943
PFA	0.7499	0.8054	0.3796	0.6894	0.8128	0.3792	0.7393	0.8315	0.3545
Elo	0.7795	0.8127	0.3652	0.7272	0.8084	0.3769	0.7384	0.8303	0.3558
M-Elo	0.7983	0.8249	0.3679	0.7318	0.7702	0.3898	0.7727	0.8288	0.3679
MV-Glicko	**0.8363**	**0.8402**	**0.3465**	**0.8227**	**0.8516**	**0.3504**	**0.8137**	**0.8549**	**0.3309**

Table 3. AUC, ACC, and RMSE for RiPPLE data sets). ↑ (↓) shows the higher (lower) is the better.

Model	InfoSys			Neuro			Pharm		
	AUC ↑	ACC ↑	RMSE ↓	AUC ↑	ACC ↑	RMSE↓	AUC ↑	ACC ↑	RMSE ↓
IRT	0.7299	0.7203	0.4302	0.7249	0.7168	0.4329	0.7556	0.81840	0.3703
AFM	0.6419	0.7026	0.4474	0.6308	0.6808	0.4561	0.6484	0.8096	0.3847
PFA	0.6272	0.6981	0.4517	0.5937	0.6839	0.4615	0.6329	0.8128	0.3826
Elo	0.7252	0.7258	0.4312	0.7324	0.7203	0.4317	0.7467	0.8194	0.3609
M-Elo	0.7277	0.7184	0.4307	0.7301	0.7184	0.4412	0.7536	0.8166	0.3648
MV-Glicko	**0.7695**	**0.7433**	**0.4176**	**0.7705**	**0.7472**	**0.4158**	**0.7852**	**0.8221**	**0.3607**

are ranked as the third-best model. The similarity between IRT and Elo performance is predictable as they both use the same form of equations for the prediction of a correct response to an item by a student, and their difference lies in the procedure they follow to estimate the model parameters: IRT generally relies on maximum likelihood estimation, but Elo follows a heuristic approach for this task [20]. As it was also demonstrated by previous studies [27], PFA and AFM provide a lower predictive performance compared with IRT, except for BAlg2006 that very slightly PFA outperforms IRT and Elo. As [27] elaborates, this might be because PFA and AFM only consider the concept-level biases. However, when the number of items is significant compared with the number of concepts, it is intuitive to imagine that the items come from various levels of difficulty. So, learner models that take into account the difficulty of items can provide a higher predictive performance compared with the learner models that ignore item biases.

4.3.2 Predictive Performance on RiPPLE Data Sets. Table 3 compares the predictive performance of each of the aforementioned learner models on RiPPLE data sets. Similar to the public data sets, on all RiPPLE data sets, MV-Glicko outperforms other conventional models based on the three evaluation criteria. In contrast to the case of public data sets where M-Elo was the second-best model, for these data sets, Elo, IRT, and M-Elo represent very similar performance. This can be explained by the fact that in RiPPLE data sets, the domain consists of a limited number of concepts (7 for InfoSys, 21 for Neuro, and 14 for Pharm), suggesting an almost homogeneous domain. As such, M-Elo that accounts for students' concept-level knowledge states does not provide superior predictive performance compared to Elo and IRT that utilise only one global parameter for estimating each student's knowledge state. However, the value of any learner model that accounts for students' concept-level knowledge states goes beyond providing superior predictive performance, and there are other advantages associated with them. For example, the additional concept-level parameters of these models provide insight into the characteristics of the domain and learning processes, which in turn can be utilised to lead the adaptive behaviour of the AES [2, 21]. Also, if the learner model is opened based on the principles of OLMs, it may deliver further insights into the course-level and student-level proficiency and gaps, which can be used by instructors to provide personalised feedback to students or to rectify the design of items, while also it may yield meta-cognitive advantages for students including increased motivation and trust in the adaptive behaviour of the AES.

4.4 Sensitivity Analysis

4.4.1 Impact of ν on the predictive performance of MV-Glicko. As it is explained in Section 3.2, the parameter ν controls the reliability of MV-Glicko estimates about students' ratings with the passage of time. Larger values of ν indicate a more significant change in the reliability of MV-Glicko estimates with the passage of time and $\nu = 0$ makes the

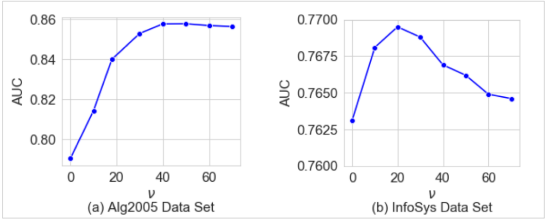

Fig. 1. Impact of different values of ν

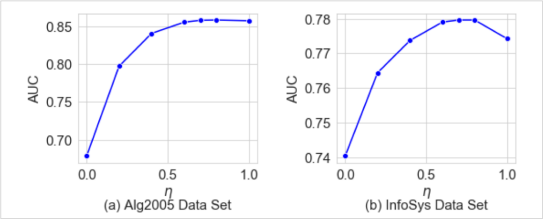

Fig. 2. Impact of different values of η

reliability of the MV-Glicko estimates independent of the passage of time. In order to gain more insight on the impact of ν on the performance of MV-Glicko, we select "Alg2005" as a sample of public data sets and "InfoSys" as a sample of RiPPLE data sets and compare the AUC of MV-Glicko for different values of ν in both data sets. Fig. 1-a, and Fig. 1-b compares the AUC of MV-Glicko for "Alg2005" and "InfoSys" data sets, respectively. As shown in Fig. 1-a for "Alg2005", as ν is increased from 0, the performance of MV-Glicko shows improvement and provides its best performance at $\nu = 50$. Beyond $\nu = 50$, the performance of MV-Glicko shows slight degradation from its maximum value. For "InfoSys" data set, by increasing the the value of ν from 0, the performance of MV-Glicko shows improvement and gets to its best performance at $\nu = 20$. However, the improvement in AUC for "Alg2005" is more considerable than "InfoSys" (+0.0675 for "Alg2005" vs. +0064 for "InfoSys"). This might be explained by the fact that with almost a similar average of the study period for students in both courses (109.9 days for "Alg2005" and 91 days for "InfoSys"), the number of knowledge components associated with learning items of "Alg2005" were huge compared with "InfoSys" (112 knowledge components for VS. "7" for "InfoSys"). As such the average of the time interval between two interactions associated with the same knowledge component for students in "InfoSys" was significantly smaller compared with "Alg2005" (3.36 days for "Alg2005" VS. 1.4 for "InfoSys"), which intrinsically reduces the impact of ν on the updates of students' knowledge states. The findings of this experiment suggest that the choice of ν is very much data-set dependent, and it is required to be carefully selected.

4.4.2 Impact of η on the predictive performance of MV-Glicko. As explained in Section 3.2, the parameter η controls the amount of penalty applied to the MV-Glicko estimates of students' ratings if they submit a wrong answer for an item. In MV-Glicko formulation, $\eta = 0$ means that no penalty is applied to students' ratings and $\eta = 1$ means that a wrong answer is penalised as much as submitting a correct answer is rewarded. There are two important justifications for considering different constants for penalising and rewarding students' knowledge states: (1) regardless of the outcome, acquisition of knowledge happens when practising a learning item [21], (2) penalising a wrong submission as much as rewarding a correct submission might discourage students' engagement with the AES [2]. Here, to gain insight on the impact of η on the performance of MV-Glicko, we again select "Alg2005" and "InfoSys" as a sample data sets and compare the AUC of MV-Glicko for different values of η ranging from 0 to 1 with the step value of 0.2. As it is shown in Fig. 2, for both data sets, setting the value of η in the range of $0 - 0.5$ provides a considerably lower performance compared with bigger values of η. On the other hand, MV-Glicko provides its best performance for the values of η in the range of $0.6 - 0.8$. For $\eta = 1$, the predictive performance of MV-Glicko is slightly degraded compared with $\eta = 0.8$.

5 CONCLUSIONS AND FUTURE WORKS

In this paper, we presented a new learner model called Multi-Concept Glicko rating system (MV-Glicko) for tracking students' knowledge state and estimating the difficulty of learning items within adaptive educational systems. Unlike state-of-the-art Elo-based learner models that are not generally sensitive to lag time between consecutive interactions of a system, MV-Glicko is sensitive to the lag time and models it as a parameter that captures the confidence of the model

about the estimated mastery for the student. The results of our experiments on three public data sets and three data sets obtained from an authentic adaptive learning platform called RiPPLE provide empirical evidence that MV-Glicko outperforms other conventional models in predicting students' performance. There are several interesting directions to be followed in the future. Given that MV-Glicko can be easily adjusted for different situations, one interesting direction would be to follow the recent works of [1] and [29] to extend the proposed model for the learning systems where students may engage with a diverse set of graded or non-graded learning activities in the system such as attempting multiple choice questions or open-ended questions, watching video lectures, participating in discussions, or engaging with other activities in the system such as creating learning items or moderating learning items. In additions, discussions are underway with the development team of RiPPLE for integrating the proposed learner model into their platform so that we can practically evaluate its fit in a real-time setting and conduct live user studies.

REFERENCES

[1] Solmaz Abdi, Hassan Khosravi, and Shazia Sadiq. 2020. Modelling Learners in Crowdsourcing Educational Systems. In *International Conference on Artificial Intelligence in Education*. Springer, 3–9.

[2] Solmaz Abdi, Hassan Khosravi, Shazia Sadiq, and Dragan Gasevic. 2019. A multivariate Elo-based learner model for adaptive educational systems. *arXiv preprint arXiv:1910.12581* (2019).

[3] Solmaz Abdi, Hassan Khosravi, Shazia Sadiq, and Dragan Gasevic. 2020. Complementing Educational Recommender Systems with Open Learner Models. In *Proceedings of the Tenth International Conference on Learning Analytics Knowledge*. Association for Computing Machinery, New York, NY, USA, 360–365.

[4] Susan Bull and Judy Kay. 2010. Open learner models. In *Advances in intelligent tutoring systems*. Springer, 301–322.

[5] Hao Cen, Kenneth Koedinger, and Brian Junker. 2006. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *International Conference on Intelligent Tutoring Systems*. Springer, 164–175.

[6] Benoît Choffin, Fabrice Popineau, Yolaine Bourda, and Jill-Jênn Vie. 2019. DAS3H: Modeling Student Learning and Forgetting for Optimally Scheduling Distributed Practice of Skills. *arXiv preprint arXiv:1905.06873* (2019).

[7] Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction* 4, 4 (1994), 253–278.

[8] Michel C Desmarais and Ryan S Baker. 2012. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction* 22, 1-2 (2012), 9–38.

[9] Philipp Doebler, Mohsen Alavash, and Carsten Giessing. 2015. Adaptive experiments with a multivariate Elo-type algorithm. *Behavior research methods* 47, 2 (2015), 384–394.

[10] Hermann Ebbinghaus. 2013. Memory: A contribution to experimental psychology. *Annals of neurosciences* 20, 4 (2013), 155.

[11] Mark E Glickman. [n.d.]. The glicko system. ([n. d.]).

[12] Mark E Glickman. 1999. Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 48, 3 (1999), 377–394.

[13] Hassan Khosravi, Kirsty Kitto, and Williams Joseph. 2019. RiPPLE: A Crowdsourced Adaptive Platform for Recommendation of Learning Activities. *Journal of Learning Analytics* 6, 3 (2019), 91–105.

[14] Hassan Khosravi, Shazia Sadiq, and Dragan Gasevic. 2020. Development and adoption of an adaptive learning system: Reflections and lessons learned. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 58–64.

[15] Sharon Klinkenberg, Marthe Straatemeier, and Han LJ van der Maas. 2011. Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education* 57, 2 (2011), 1813–1824.

[16] Kenneth R Koedinger, Albert T Corbett, and Charles Perfetti. 2012. The Knowledge-Learning-Instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive science* 36, 5 (2012), 757–798.

[17] Koki Nagatani, Qian Zhang, Masahiro Sato, Yan-Ying Chen, Francine Chen, and Tomoko Ohkuma. 2019. Augmenting Knowledge Tracing by Considering Forgetting Behavior. In *The World Wide Web Conference*. Association for Computing Machinery, New York, NY, USA, 3101–3107. https://doi.org/10.1145/3308558.3313565

[18] Alexandros Paramythis and Susanne Loidl-Reisinger. 2003. Adaptive learning environments and e-learning standards. In *Second european conference on e-learning*, Vol. 1. 369–379.

[19] Philip I Pavlik Jr, Hao Cen, and Kenneth R Koedinger. 2009. Performance Factors Analysis–A New Alternative to Knowledge Tracing. *Online Submission* (2009).

[20] Radek Pelánek. 2016. Applications of the Elo rating system in adaptive educational systems. *Computers & Education* 98 (2016), 169–179.

[21] Radek Pelánek, Jan Papoušek, Jiří Řihák, Vít Stanislav, and Juraj Nižnan. 2017. Elo-based learner modeling for the adaptive practice of facts. *User Modeling and User-Adapted Interaction* 27, 1 (2017), 89–118.

[22] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in neural information processing systems*. 505–513.

[23] Yumeng Qiu, Yingmei Qi, Hanyuan Lu, Zachary A Pardos, and Neil T Heffernan. [n.d.]. Does Time Matter? Modeling the Effect of Time with Bayesian Knowledge Tracing.

[24] Rachel Reddick. 2019. Using a Glicko-Based Algorithm to Measure In-Course Learning. *International Educational Data Mining Society* (2019).

[25] J Stamper, A Niculescu-Mizilm, S Ritter, GJ Gordon, and KR Koedinger. 2010. Data set from KDD Cup 2010 educational data mining challenge.

[26] Wim J van der Linden and Ronald K Hambleton. 2013. *Handbook of modern item response theory*. Springer Science & Business Media.

[27] Jill-Jênn Vie and Hisashi Kashima. 2019. Knowledge tracing machines: Factorization machines for knowledge tracing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 750–757.

[28] Kelly Wauters, Piet Desmet, and Wim Van Noortgate. 2010. Monitoring learners' proficiency: weight adaptation in the elo rating system. In *Educational Data Mining 2011*.

[29] Siqian Zhao, Chunpai Wang, and Shaghayegh Sahebi. 2020. Modeling Knowledge Acquisition from Multiple Learning Resource Types. *arXiv preprint arXiv:2006.13390* (2020).