IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 30 January 2023, accepted 14 February 2023, date of publication 27 February 2023, date of current version 2 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3249286

RESEARCH ARTICLE

Modeling Learners to Early Predict Their Performance in Educational Computer Games

DANIAL HOOSHYAR[1], NOUR EL MAWAS[2], MARCELO MILRAD[3], (Member, IEEE), AND YEONGWOOK YANG[4]

[1]School of Digital Technologies, Tallinn University, 10120 Tallinn, Estonia
[2]CIREL, University of Lille, 59000 Lille, France
[3]Department of Computer Science and Media Technology, Linnaeus University, 35252 Växjö, Sweden
[4]Division of Computer Engineering, Hanshin University, Osan 18101, South Korea

Corresponding author: Yeongwook Yang (yeongwook.yang@gmail.com)

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIT) under Grant 2021R1C1C2004868.

ABSTRACT Data mining approaches have proven to be successful in improving learners´interaction with educational computer games. Despite the potential of predictive modelling in providing timely adaptive learning and gameplay experience, there is a lack of research on the early prediction of learners' performance in educational games. In this research, we propose an early predictive modelling approach, called GameEPM, to estimate learners' final scores in an educational game for promoting computational thinking. Specifically, the GameEPM approach models the sequence of learners' actions and then uses a limited sequence of the actions to predict the final score of the game for each learner. The findings from our initial trials show that our approach can accurately and robustly estimate the learners' performance at the early stages of the game. Using less than 50% of learners' action sequences, the cross-validated deep learning model achieves a squared correlation higher than 0.8 with a relative error of less than 8%, outperforming a range of regression models like linear regression, random forest, neural networks, and support vector machines. An additional experiment showed that the validated deep learning model can also achieve high performance while tested on an independent game dataset, showing its applicability and robustness in real-world cases. Comparing the results with traditional machine learning methods revealed that, in the validation and application phases, up to 0.30 and 0.35 R^2 gain is achieved in favor of the deep learning model, respectively. Finally, we found that while the lengths of action sequences influence the predictive power of the traditional machine learning methods, this effect is not substantial in the deep learning model.

INDEX TERMS Early performance prediction, learner model, educational games, computational thinking, deep learning.

I. INTRODUCTION

Equipping educational computer games (hereafter called educational games) with adaptivity has shown to be effective in improving learners' engagement, motivation, and learning gain in a wide range of fields (e.g., [1], [2], [3], [4], [5]). Such educational games provide customization of learning and/or gameplay to match the individual needs and interests of learners while playing. There are different ways to achieve adaptive learning in educational games, for instance, through knowledge tracing that aims to model learners' knowledge over time and accordingly predict their learning performance (e.g., [6], [7], [8]). Prediction of learners' performance may help educators to provide learners with more appropriate learning materials, timely and personalized feedback, recommend tutorials and personalized learning paths, etc. (e.g., [9], [10]).

Several research efforts have been carried out revolving around the prediction of learners' performance in educational games. For instance, Ghali et al. [6] developed a

The associate editor coordinating the review of this manuscript and approving it for publication was Dongxiao Yu.

machine learning-based approach to model learners' learning behaviour and accordingly predict whether they succeed or fail in game episodes of an educational game called LewiSpace. Additionally, Gagnon et al. [11] used raw clickstream in-game data to model players quitting the game at each level. While successful, such predictive modelling approaches suffer from several shortcomings. For instance, they mostly assume that players' actions during gameplay are independent and neglect considering sequences of players' actions. Also, they often rely on traditional machine learning methods and overlook the prediction power of deep learning models. Most importantly, they sometimes overlook considering learners' in-game data for *early* prediction of their performance. As stated by Macfadyen and Dawson [12], early prediction and timely intervention is the main goal of predictive modelling. From an educational perspective, even though the prediction of learners' performance could have the potential to improve different types of educational games, in practice such predictive models are unable to make a stable and timely prediction at the early stages of games. If learners are alerted about their performance after the game episode is finished (in other words, if the time series factor is not considered), how could educators or the game itself offer individualized interventions to avoid failure? Predicting the performance of learners early on in a game episode could support both the learning and teaching process. For instance, it could offer timely interventions like feedback, hints, and tutorials, adapting the instruction methods, etc. Such strategies support providing special attention at the early stages and minimizing unproductive practices.

Recently, there have been some attempts to early predict learners' performance in educational games. For example, Lee-Cultura et al. [13] applied machine learning methods to multi-modal data (e.g., eye trackers and wristbands) to predict learners' academic performance in an arithmetic operations game. Wiggins et al. [14] developed an approach that uses the facial expressions of learners to predict engagement and perceived mental demand. Milošević et al. [15] proposed an early churn prediction approach that can predict early one-day churn with decent performance. Moreover, Hooshyar et al. [8] developed a deep knowledge tracing approach to estimate learners' knowledge state over time and accordingly predicted their performance in different game tasks. Despite the potential of the proposed early predictive modelling approaches, they cannot mainly deal with the regression task of early prediction of performance and often revolve around binary classification tasks, ignore employing deep learning models (except the approach proposed by Hooshyar et al. [8]) and comparing their performances with traditional machine learning methods, and overlook investigating the effect of various sequences of learners' actions on the prediction power of their approaches.

In this specific research, by considering learners' sequential behaviour in educational games, we aim to build on the existing works by proposing an approach called GameEPM that can predict learners' final performance at the early stages of the game. To this aim, we set the following research questions:

- *How accurately can we predict learners' final performance at the early stages of educational games?*
- *How is the performance of a deep learning model compared to traditional machine learning methods in tackling the regression task of early prediction of learners' performance?*
- *What are the effects of various sequences of learners' actions on the prediction power of different regression models?*

To answer the research questions, we investigate the feasibility and performance of the GameEEP approach by applying it to data from the AutoThinking game which is an educational game for promoting computational thinking [5]. This enables the game to continuously estimate the performance of learners and offers adaptivity in both learning and gameplay. For instance, it can adaptively regulate or activate the feedback system, hints, and tutorials employed in the game. Additionally, early prediction can help to monitor learners' actions in the games and flag at-risk learners, potentially offering timely warnings, as well as facilitating the design of adaptive non-player characters (NPCs). Contributions of our research are manifold: i) a novel and validated predictive modelling approach that considers learners' action sequences to predict their final scores at the early stages of the game; ii) comparing a state-of-the-art prediction model (i.e., deep learning) with several traditional machine learning methods for tackling the regression task of early prediction of learners' scores in educational games; iii) investigating the applicability of the validated models by testing them on an independent educational game dataset; iv) investigating the effect of various sequences of learners' actions on the prediction power of different regression models.

The structure of this article is as follows: section two presents related research efforts in this field while section three deals with materials and methods. Thereafter, section four illustrates the experimental evaluation proceeded by section five which revolves around results and analysis. Finally, the paper ends with section six in which we present and discuss our conclusions.

II. RELATED RESEARCH

There has been substantial research spinning around the application of data mining approaches in educational contexts aimed at better understanding learners' behaviour and accordingly providing individualized interventions (this area of research is called EDM or Educational Data Mining). Bienkowski et al. [16] reported use cases of data mining techniques for different purposes such as predictive modelling to facilitate learning. Usually, most predictive modelling employs stealth assessment strategies to measure different aspects of the learning process; in other words, they take into account learners' sequence of actions while interacting with learning platforms and accordingly estimate their skills, knowledge, motivation, performance, etc. [17].

Tan and Shao [18] predicted student dropout in e-learning platforms using machine learning methods. Patil et al. [19] developed an approach to predicting a learner's grade point average. Iqbal et al. [20] put forward a machine learning-based approach to predict students' grades. Hooshyar et al. [10] modelled learners' submission behaviour, and through their procrastination behaviour predicted their performance in online learning.

Bujang et al. [21] proposed a data mining approach for the prediction of learners' final grades according to the results of their previous examinations. Liu et al. [22] developed a deep knowledge tracing approach that considers learners' previous interactions with exams/tasks to model their knowledge over time and accordingly predict their future performance in the upcoming tasks. Alamri and Alharbi [23] and Hellas et al. [9] offer systematic reviews of student performance prediction using machine learning techniques. Similarly, to improve the effectiveness of educational games, there have been many studies that deal with mining data from (educational) games to predict players' performance, success, dropout, and many more (e.g., [8], [24], [25], [26]). Many researchers have indicated that educational games can reach their full educational potential if they consider learners' individual needs and skills and accordingly offer real-time feedback, in-game interventions, task difficulty, and learning content adaptation (e.g., [27], [28], [29]). In this regard, Ghali et al. [6] collected and processed learners' interaction with an educational game called LewiSpace for teaching drawing Lewis diagrams. In their research, the authors used personality traits and different physiological data (from different sources like Electroencephalography, eye tracker, and facial expression recognition) to predict learners' performance during gameplay. According to their findings, a logistic regression model could successfully detect learners with difficulties during gameplay and thus provide them with help and guidance to understand the lesson better. Wiggins et al. [14] developed an approach that observes early interactions between learners and the tutorial of an educational game and predicts learners perceived mental demand and engagement at end of the game episode. According to their findings, facial indicators of surprise were important for predicting engagement and mental demand and they hold great potential for providing effective interventions. Milošević et al. [15] proposed an early game churn prediction approach that considers a player to have churned if she/he has not returned to the game within 14 days. Accordingly, they used sequences of players' action data and employed five traditional machine learning techniques like decision trees and logistic regression for the binary classification task. Their findings revealed that their proposed approach can to a certain extent early predict players' churn. They also mentioned that including more game-specific features has the potential to improve the performance of their predictive modelling approach. Gagnon et al. [11] used raw clickstream data from two educational games—i.e., the Crystal Cave and Wave Combinator—to predict quitting and the performance of learners during gameplay. The authors used data samples of 1,254 and 5,308 from anonymous internet players during the fall of 2018 and after pre-processing and feature engineering, they developed predictive models. Their findings revealed that logistic regression could successfully predict learners' quitting and performance, outperforming other machine-learning methods. Roberto et al. [30] put forward an approach that applies neural networks to learners' in-game data (from an educational game called Ratio Rancher) to predict their post-test scores. Based on their findings, while data on learners' actions during gameplay could predict their post-test score with an accuracy of almost 62%, coupling the in-game data with their pre-test scores could increase the accuracy of the model to 75%.

In a different vein, Lee-Cultura et al. [13] proposed a machine learning-based approach that benefits from multi-modal data such as Empatica E4 Wristband, eye tracker, and Kinect Skeleton to early predict learners' academic performance in an arithmetic operations game. Their findings showed that an ensemble learner combining gaze and physiological data could predict learners' performance at early stages. Moreover, Hooshyar et al. [8] developed a novel predictive modelling approach that combines domain knowledge with advanced machine learning methods to estimate learners' success in the next game tasks. The approach called GameDKT consumes learners' in-game data from an educational game for fostering computational thinking. Their findings revealed that learners' knowledge could be modelled or traced over time through their past activities, facilitating the prediction of their performance in the upcoming game tasks. A convolutional neural network with an accuracy of 85% was found to be the best for the prediction task.

As it is apparent, there has not been much research considering in-game data of learners to early predict their final scores in educational game episodes. Moreover, the existing related works mainly ignore: i) tackling the regression problem of early prediction of learners' final score (rather deal with binary classification tasks); ii) employing deep learning models and comparing their performance with traditional machine learning methods; iii) investigating the effect of various sequences of learners' actions on the prediction power of different regression models; iv) investigating the applicability of the validated models by testing them on an independent educational game dataset. Addressing the named challenges contribute to the literature and facilitates offering proactive feedback like hints, opportunities for reflection throughout the learning process, task difficulty and learning content adaptation, and many more. This research aims to bridge the gap by developing an early predictive model for estimating learners' final scores at the very early stages of the game.

III. MATERIALS AND METHODS

To predict learners' performance at early stages, we proposed an approach called GameEPM benefiting from a series of time windows related to learners' sequence of actions in educational games. In the subsequent sections, a brief context of the early prediction task and a description of the GameEPM approach are presented.

A. CONTEXT ON THE EARLY PREDICTION

The AutoThinking[1] game includes three levels where players should, in a role of a mouse, develop different types of strategies or solutions to-–collect as much score as possible, and escape from two cats in the maze—complete or win the level. In this context, learners can develop up to 20 solutions to escape the mouse from NPCs and collect scores. Learners can develop low- or high-quality solutions during the gameplay (e.g., high-quality solutions consider the use of conditional and loop concepts, or pattern recognition and generalization). For this research, we use the learner's sequence of actions or solution submissions from the third level of the game. In each game episode, our task is to predict the final score of the learners using a limited sequence of their actions/attempts over time (e.g., using 30% of their action sequences to predict the final score of the game episode). For the early prediction task, we identify the learner's sequence of actions in each time window. Specifically, for each learner, we first create varying time windows of the game data. This means considering a sequence of learners' actions and a tuneable maximum solutions size τ, we extract a feature vector based on the information on learners' solutions and actions from each trial solution at intervals of ($\tau \times p$) (where p is the percentage of the player's sequence of actions). Thereafter, from each time window, we create a collection of different solutions and actions in each game episode. Once the time windows are generated from a sequence of learners' actions, their final score at the end of each game episode is used to signal their final performance.

B. THE PROPOSED GameEPM APPROACH

Algorithm 1 illustrates the step-by-step process of the GameEPM approach. Additionally, Fig. 1 presents a general overview of the approach.

As shown in the figure, in the first step, we first collect and pre-process learners' sequence of solution submissions. For this purpose, a combination of features that directly or indirectly deal with learners' final scores was used. For example, this can include the number of collected objects in a game, the score gained for doing a specific action, the game task IDs, learner IDs, and so on.

We filtered out game data belonging to those learners who had not completed the game episode. The reason for such data exclusion is that we needed to use the learner's final score to signal their final performance. Thereafter, we normalized our data using a range transformation with min and max of 0 and 1, respectively. To remove noise from our dataset, we implemented the outlier detection method of local outlier factors using Euclidean distance. Subsequently, we converted the sequence of learners' actions or solution submissions into varying time windows (sequence of input vectors) and labelled them with their final scores. More specifically, for each learner, we created time windows using 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100% of their action sequence. After creating the nonstationary time windows, we created multiple datasets for early prediction of the learner's final score. Because some learners completed the game with less than 20 solutions, which may have resulted in winning or losing, we used the Nearest Neighbour imputation technique to create a stationary example set for each time window (i.e., 30% to 100%). Finally, we applied a single exponential smoothing technique to reduce the absence of key information, allowing important patterns to stand out which makes the problem easier to learn by regression models.

In the second step, after creating the sequence of fixed length input vectors, we employed and trained eight traditional machine learning algorithms (i.e., neural networks, linear regression, k-nearest neighbours, support vector machines, random forest, decision tree, generalized linear model, and gradient boosted model) and a multilayer perceptron (hereafter called deep learning) to encode the latent knowledge states of learners. To fine-tune the parameters of the deep learning model, we used an evolutionary optimisation method, i.e., a Genetic algorithm, over the combinations of parameters. The deep learning model includes two fully connected hidden layers with 50 neurons each.

Finally, we used 10-fold cross-validation with shuffled sampling to validate the performance of our models, was used. Additionally, we used independent test sets to show the practicality, stability, and robustness of the validated models. To select the best regression model among the nine, we employed a t-test followed by an analysis of variance (ANOVA) to compare the models through performance measures.

Algorithm 1 Early Prediction of Learners' Performance

Input: Dataset D
Output: Regression results among R
Initialize i =0
$X(A)$ = Feature Selection(A)
X = Normalization(X)
X = Outlier detection(X)
for i in N //N is the number of learners
  u_i =Time windowing $T(t_\tau, t_{\tau\times p})$ of each u in X
U = KNN imputation for missing values(U)
for i in N // Exponential Smoothing
  for j in m
    $t'^{i}_{j\times 1} = \alpha t^{i}_{j} + (1-\alpha)\, t'^{i}_{j}$
Apply regression methods using U
Compare the performance of regression methods through cross-validation and independent dataset
Select the algorithm with the highest performance
Use the algorithm for early prediction of players' performance

[1] https://www.youtube.com/watch?v=O3K6G0i1jYU

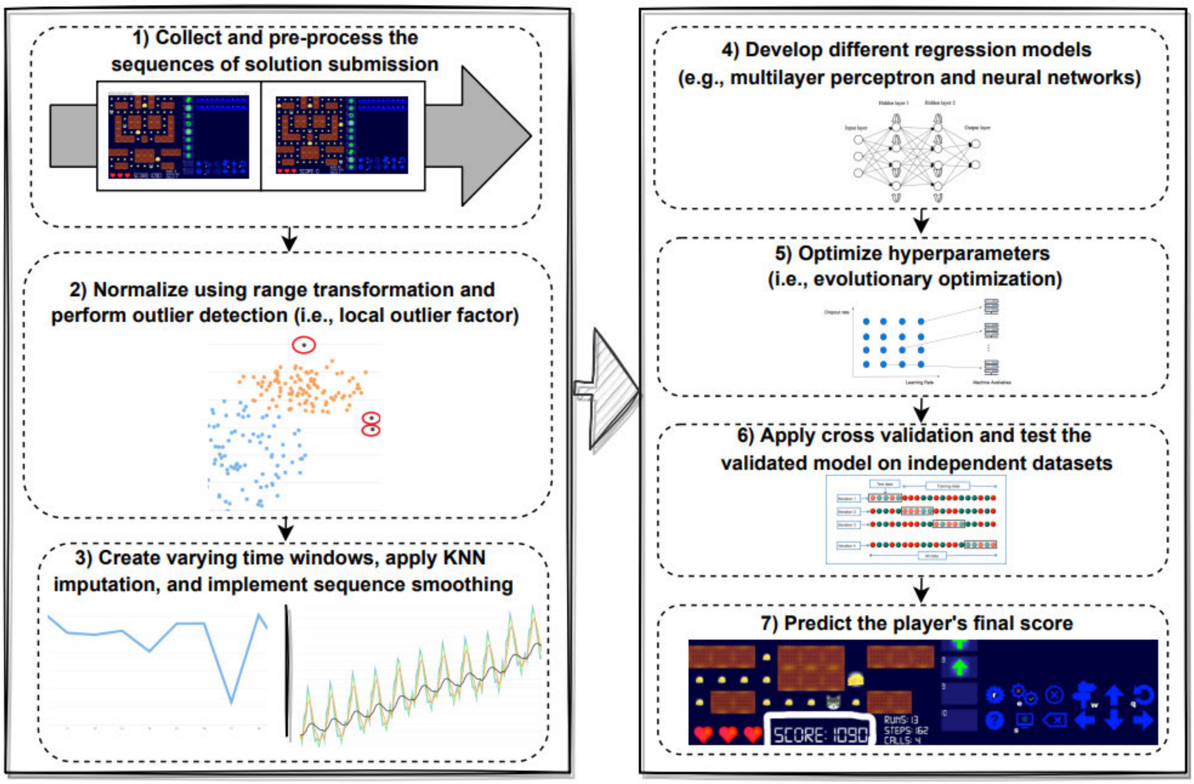


FIGURE 1. The overall proposed framework of the GameEPM approach.

TABLE 1. Notations.

Notation	Explanation
U, u	A set of Users and a specific user, $U = \{u_1, u_2, u_3, \dots, u_n\}$, $u_n = (t_1, t_2, \dots, t_\tau)$
T, t	A set of trials and a specific trial, $T = \{t_1^u, t_2^u, t_3^u, \dots, t_m^u\}$, $t_m = (a_1, a_2, a_3, \dots, a_l)$
A, a	A set of features and a specific feature, $A = \{a_1, a_2, a_3, \dots, a_l\}$
R	The regression methods

IV. RESULTS AND FINDINGS

A. DATASETS

In this research, we used log data of 427 learners from five countries (Estonia, France, South Korea, Taiwan, and South Africa) who have played the third level of the AutoThinking game. Learners had different backgrounds and ages. The dataset includes up to 6199 examples or solutions collected between December 2019 and April 2022. While some part of the data was collected during experimental studies (e.g., [31], [32]), some learners played the game on their own. To complete the game, learners developed 20 or fewer solutions, depending on the game episode' conditions.

During the gameplay, various interaction of learners is logged, for instance, the position of mouse and NPCs, task IDs, number of small and big cheese eaten, loop and conditional usage, arrow and function usage, debug and simulation usage, number of times that help was requested, frequency of given feedback and hints, frequency of hitting walls, the estimate of learner's overall knowledge of computational thinking or the quality of the developed solutions (inferred from the Bayesian network decision-making algorithm used in the game), etc. To generate the Bayesian estimates, the game considers multiple factors such as the potential of gaining scores, the risk of being caught by the NPCs, and the quality of computational thinking skills and concepts used in solutions. Further information on the decision-making process can be found in Hooshyar et al. [33].

Of all the features, based on the design and mechanism used in the game, we opted for those that directly or indirectly deal with learners' final scores, for instance, the number of big and small cheeses eaten, score gained for actions done in a single solution, the game task IDs, learner IDs, number of times bumping into walls, overall score gained for one solution, frequency of using loop and conditional concepts, and the final score of the game episode.

To validate the GameEPM approach, by splitting the original dataset using shuffled sampling, we used data from 309 learners for the validation phase and an independent 30 learners (560 solutions) for the application phase. Additionally, to test the capability of the proposed approach in the early prediction of the learner's final performance, we developed eight different versions of each dataset (for cross-validation and application datasets), namely 30% to

TABLE 2. Squared correlation, RMSE, and relative error of the models for different sequence lengths using cross-validation.

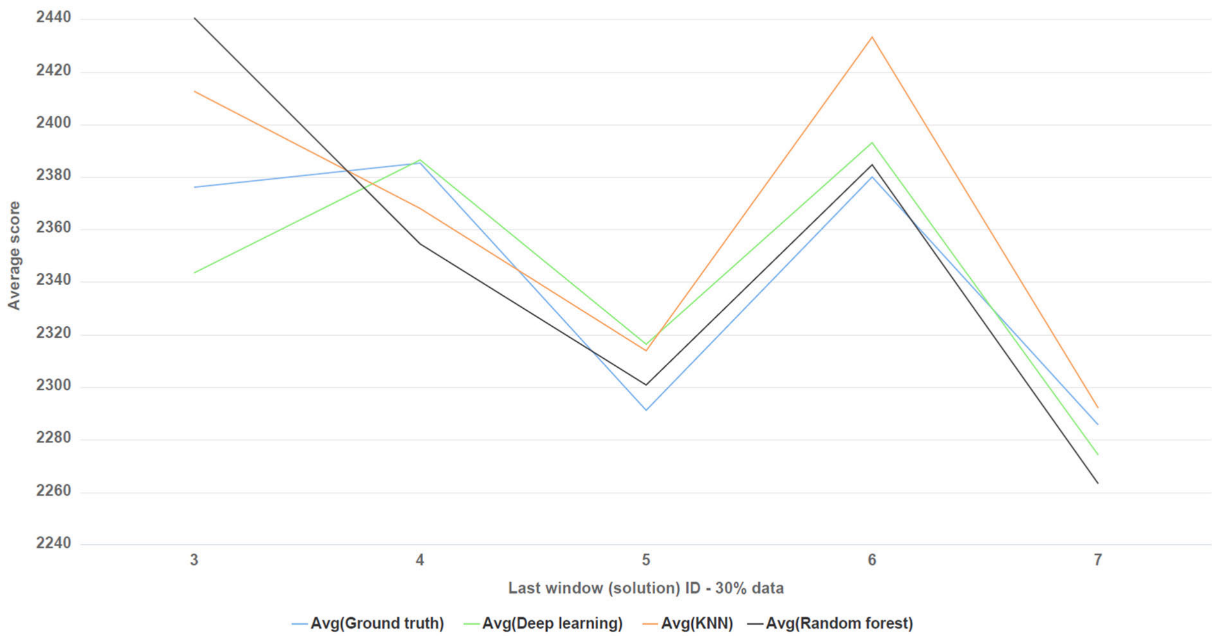
Models	Metrics	Sequence lengths							
		30% of solutions	40% of solutions	50% of solutions	60% of solutions	70% of solutions	80% of solutions	90% of solutions	100% of solutions
Deep learning	R^2	0.754 +/- 0.075	0.802 +/- 0.067	0.816 +/- 0.085	0.843 +/- 0.048	0.847 +/- 0.074	0.828 +/- 0.069	0.847 +/- 0.048	0.828 +/- 0.092
	RMSE	233.033 +/- 29.351	198.334 +/- 27.602	190.904 +/- 33.392	179.961 +/- 29.773	176.095 +/- 30.370	183.255 +/- 21.532	178.085 +/- 27.726	184.623 +/- 40.303
	RE	9.27% +/- 1.67%	7.67% +/- 1.71%	7.45% +/- 1.98%	6.96% +/- 1.69%	6.62% +/- 1.71%	7.06% +/- 1.48%	7.03% +/- 1.81%	6.61% +/- 1.27%
Neural network	R^2	0.614 +/- 0.142	0.732 +/- 0.131	0.706 +/- 0.148	0.743 +/- 0.123	0.733 +/- 0.150	0.800 +/- 0.072	0.788 +/- 0.112	0.741 +/- 0.144
	RMSE	342.305 +/- 69.014	298.519 +/- 112.099	291.842 +/- 57.053	255.592 +/- 54.248	255.027 +/- 80.185	304.333 +/- 84.673	254.123 +/- 54.096	282.604 +/- 67.720
	RE	13.04% +/- 3.30%	11.39% +/- 5.29%	10.72% +/- 2.48%	9.70% +/- 2.61%	9.19% +/- 3.00%	11.28% +/- 4.24%	10.26% +/- 2.67%	10.96% +/- 3.29%
Linear regression	R^2	0.449 +/- 0.138	0.521 +/- 0.220	0.522 +/- 0.201	0.650 +/- 0.096	0.682 +/- 0.136	0.683 +/- 0.134	0.816 +/- 0.081	0.727 +/- 0.198
	RMSE	336.414 +/- 54.451	316.261 +/- 51.630	323.669 +/- 53.380	285.554 +/- 46.168	257.638 +/- 46.706	262.512 +/- 43.519	192.276 +/- 17.088	235.511 +/- 29.252
	RE	13.67% +/- 3.80%	12.52% +/- 4.74%	12.33% +/- 3.77%	10.73% +/- 1.79%	9.63% +/- 2.25%	9.97% +/- 1.72%	7.15% +/- 1.15%	8.46% +/- 1.19%
KNN	R^2	0.613 +/- 0.162	0.697 +/- 0.102	0.682 +/- 0.102	0.638 +/- 0.145	0.675 +/- 0.135	0.692 +/- 0.135	0.678 +/- 0.222	0.713 +/- 0.139
	RMSE	298.514 +/- 55.150	272.616 +/- 49.726	270.208 +/- 50.298	278.938 +/- 60.136	271.802 +/- 54.984	264.880 +/- 50.152	256.098 +/- 43.075	257.883 +/- 37.561
	RE	12.40% +/- 4.84%	11.21% +/- 3.45%	10.98% +/- 3.15%	11.48% +/- 4.83%	11.36% +/- 5.06%	10.85% +/- 3.59%	10.49% +/- 4.11%	10.44% +/- 2.94%
SVM	R^2	0.563 +/- 0.149	0.568 +/- 0.106	0.562 +/- 0.112	0.589 +/- 0.122	0.552 +/- 0.139	0.582 +/- 0.125	0.555 +/- 0.136	0.563 +/- 0.132
	RMSE	442.099 +/- 63.203	434.204 +/- 119.264	430.184 +/- 119.545	432.534 +/- 117.357	430.907 +/- 120.645	435.041 +/- 118.102	432.570 +/- 119.651	432.458 +/- 119.632
	RE	19.14% +/- 6.99%	19.29% +/- 8.50%	19.11% +/- 8.36%	19.19% +/- 8.28%	19.08% +/- 8.39%	19.27% +/- 8.31%	19.13% +/- 8.28%	19.10% +/- 8.29%
Random forest	R^2	0.724 +/- 0.112	0.769 +/- 0.117	0.794 +/- 0.073	0.773 +/- 0.106	0.746 +/- 0.119	0.797 +/- 0.086	0.786 +/- 0.109	0.779 +/- 0.114
	RMSE	246.434 +/- 55.674	234.865 +/- 50.381	230.210 +/- 43.147	229.256 +/- 64.847	230.184 +/- 44.297	227.160 +/- 46.718	232.280 +/- 48.563	223.849 +/- 64.388
	RE	10.04% +/- 4.43%	9.60% +/- 3.02%	9.48% +/- 3.14%	9.56% +/- 5.35%	9.31% +/- 3.02%	9.21% +/- 2.97%	9.52% +/- 3.31%	9.26% +/- 4.71%
Decision tree	R^2	0.525 +/- 0.190	0.522 +/- 0.140	0.536 +/- 0.199	0.571 +/- 0.160	0.493 +/- 0.208	0.507 +/- 0.202	0.459 +/- 0.235	0.470 +/- 0.220
	RMSE	346.609 +/- 82.972	323.618 +/- 65.895	309.773 +/- 47.904	318.239 +/- 60.633	329.893 +/- 43.581	330.276 +/- 52.237	336.749 +/- 79.351	326.298 +/- 51.892
	RE	13.57% +/- 4.61%	12.69% +/- 4.30%	11.63% +/- 3.20%	11.66% +/- 3.03%	12.43% +/- 2.70%	12.33% +/- 2.73%	12.73% +/- 3.29%	12.35% +/- 1.95%
Generalized linear model	R^2	0.601 +/- 0.146	0.653 +/- 0.170	0.682 +/- 0.108	0.710 +/- 0.073	0.681 +/- 0.130	0.675 +/- 0.183	0.687 +/- 0.112	0.661 +/- 0.197
	RMSE	399.394 +/- 56.542	382.994 +/- 94.938	380.340 +/- 33.066	370.738 +/- 77.336	366.390 +/- 45.704	357.650 +/- 53.441	352.773 +/- 76.345	347.540 +/- 67.296
	RE	17.15% +/- 5.98%	16.84% +/- 8.25%	16.31% +/- 3.49%	16.14% +/- 6.51%	15.74% +/- 5.49%	15.38% +/- 5.86%	15.34% +/- 7.52%	15.01% +/- 5.33%
Gradient boosted model	R^2	0.570 +/- 0.143	0.647 +/- 0.086	0.643 +/- 0.095	0.672 +/- 0.143	0.601 +/- 0.171	0.573 +/- 0.229	0.649 +/- 0.129	0.645 +/- 0.160
	RMSE	365.267 +/- 60.390	354.019 +/- 67.167	356.083 +/- 56.036	352.037 +/- 71.862	352.768 +/- 83.528	365.433 +/- 49.318	360.527 +/- 51.195	357.222 +/- 72.482
	RE	15.59% +/- 5.36%	15.31% +/- 6.93%	15.49% +/- 4.81%	15.40% +/- 7.72%	15.58% +/- 6.48%	15.62% +/- 5.68%	15.50% +/- 4.67%	15.55% +/- 5.53%

*Squared correlation = R^2
**Relative error = RE
***'+' and '-' denote one standard deviation calculated from the ten model accuracy.

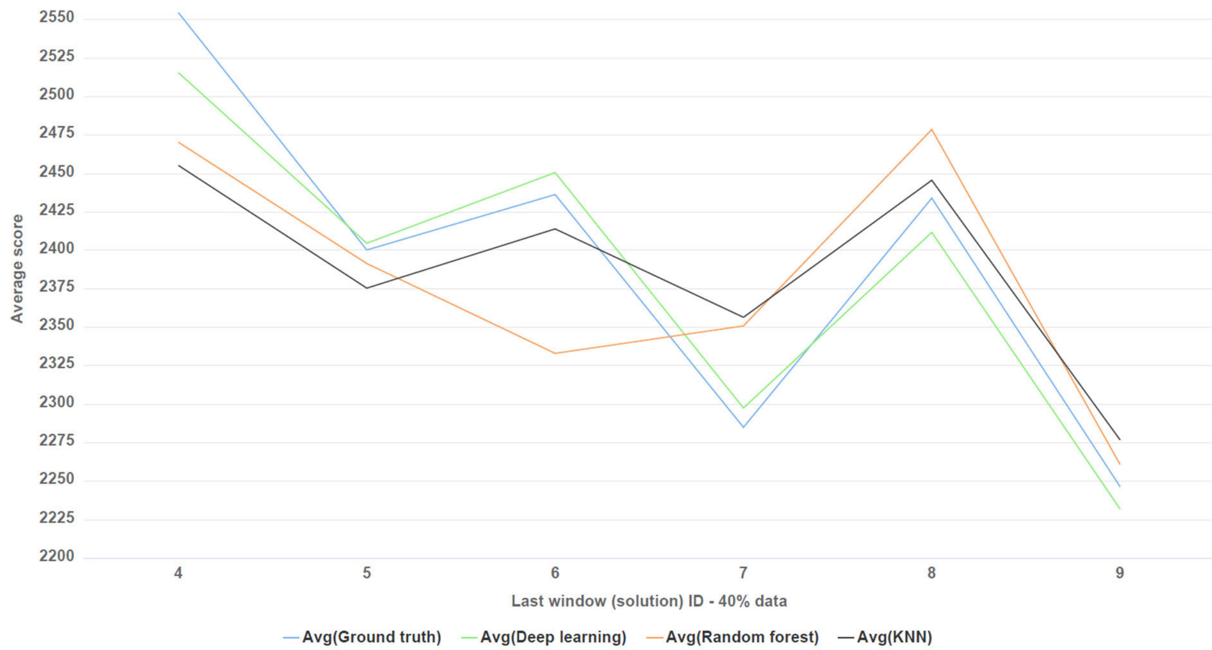
100% of the learner's action sequence. In the dataset, the min, max, and the average of the learner's final scores were 342, 3294, and 2349, respectively.

B. EXPERIMENT SETTING AND EVALUATION

We implemented the deep learning and other models on a computer with a single AMD Ryzen 5 PRO 4650U CPU and 16.0 GB memory. The deep learning model was based on a multi-layer feed-forward artificial neural network trained with stochastic gradient descent using back-propagation. The network used two hidden layers with 50 nodes per layer, a rectifier as an activation function with a learning rate of 0.005 and 10 epochs, a quadratic loss function, 1.0E-5 for L1 and 0 for L2 regularization, and Gaussian distribution.



(a)

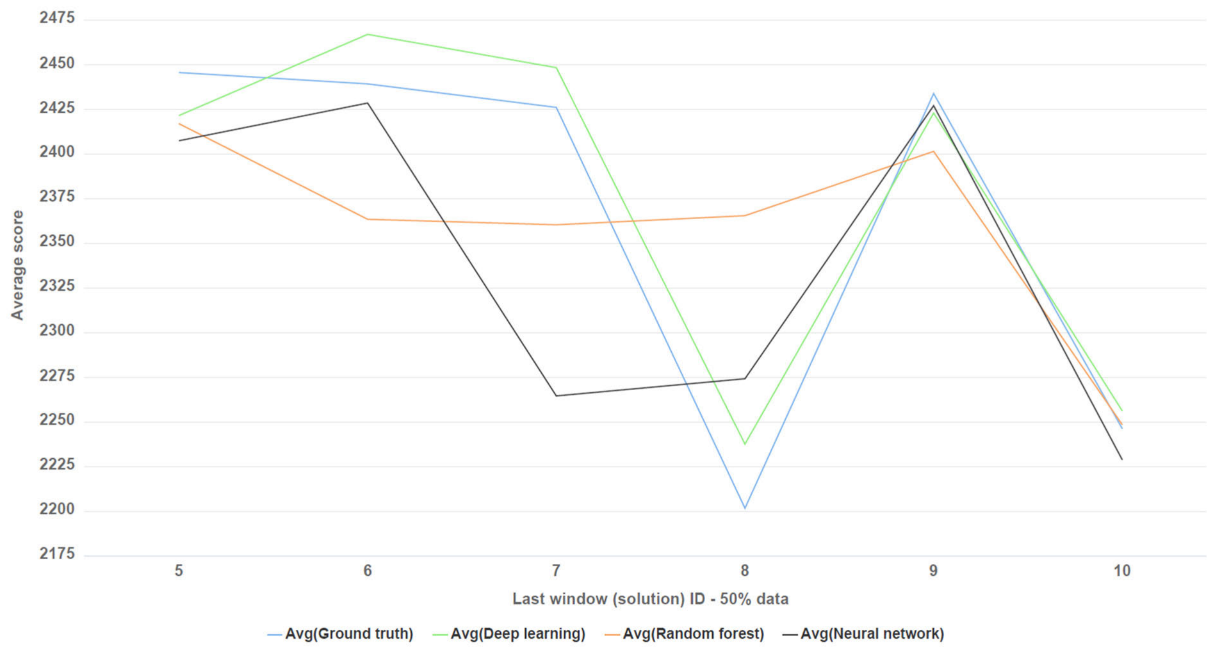


(b)

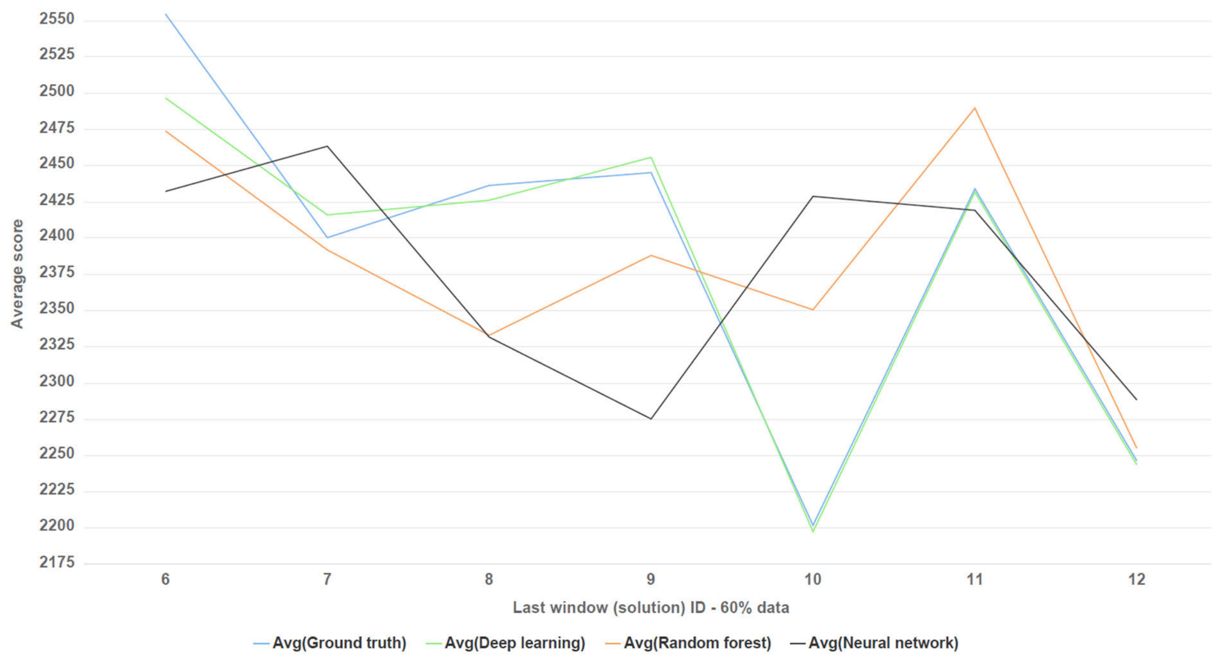
FIGURE 2. Predictions using: a) 30%, b) 40%, c) 50%, d) 60%, and e) 70% of players' solutions from the top three models along with the ground truth using cross-validation.

As some input sequences have different lengths (10 is the minimum number of solutions and 20 is the maximum solution number), we have filled up the missing time steps using KNN imputation. KNN imputation uses a model to predict missing values, where a model is developed for each feature with missing values taking other features as input values. For instance, when using 50% of the sequence of the learner's action, for those learners with only 10 solutions, we used the first five solutions and filled up the missing time steps by identifying samples closest to it and averaging those nearby points to fill in the value to create 50%-time windows for the models.

Furthermore, k-fold cross-validation was employed as it is considered among the most reliable validation methods for the future accuracy of a predictive model, especially when it comes to rather small datasets. Simply put, we employed cross-validation as it enables us to test how well our models can get trained by some data and then predict



(c)

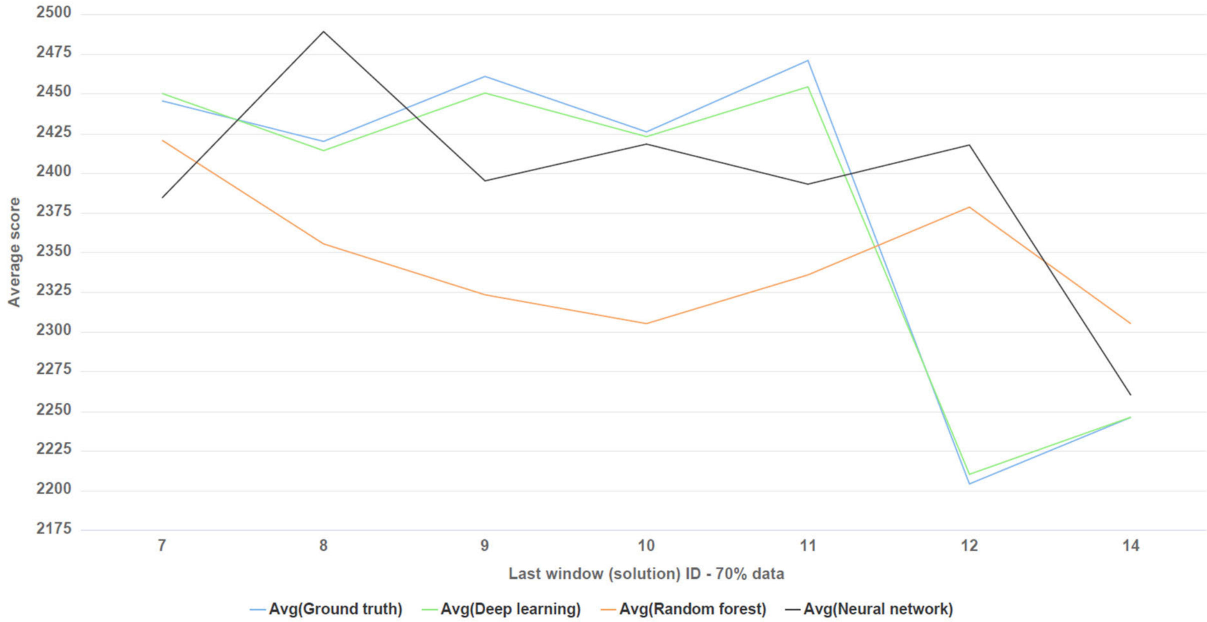


(d)

FIGURE 2. ***(Continued.)*** **Predictions using: a) 30%, b) 40%, c) 50%, d) 60%, and e) 70% of players' solutions from the top three models along with the ground truth using cross-validation.**

data it has not seen. The choice of k involves a trade-off between the error prediction's efficiency and accuracy, and k = 10 is usually preferred as it has been shown empirically to produce test error rate estimates suffering neither from excessive-high variance nor very high bias. Besides, to investigate the practicality and applicability of our models, we tested the cross-validated models on an independent game dataset of 30 learners (including 560 solutions). We compare the performance of different models on eight different sequences of learners' actions (i.e., 30% to 100% of solutions).

To measure the performance of the models, we have employed three metrics of squared correlation (R^2), root mean squared error (RMSE), and relative error (RE). RMSE shows the standard deviation of prediction errors, R^2 refers to the square of the Pearson correlation coefficient between the ground truth and predicted values, and RE reveals the magnitude of the absolute error concerning the actual size of



(e)

FIGURE 2. *(Continued.)* **Predictions using: a) 30%, b) 40%, c) 50%, d) 60%, and e) 70% of players' solutions from the top three models along with the ground truth using cross-validation.**

TABLE 3. **T-test results between the deep learning and other models.**

Best performing model	Percentage of solutions	Models							
		Neural network	*Linear regression*	*KNN*	*SVM*	*Random forest*	*Decision tree*	*Generalized linear model*	*Gradient boosted model*
Deep learning	30	.009*	.000*	.004*	.000*	.260	.001*	.000*	.000*
	40	.000*	.000*	.001*	.000*	.006*	.000*	.000*	.000*
	50	.000*	.000*	.000*	.000*	.003*	.000*	.000*	.000*
	60	.002*	.000*	.000*	.000*	.001*	.000*	.000*	.000*
	70	.003*	.000*	.000*	.000*	.000*	.000*	.000*	.000*

*shows the statistical significance at level .05
**RMSEs of the models are used to calculate the pairwise t-test

TABLE 4. **ANOVA results between the deep learning and other models.**

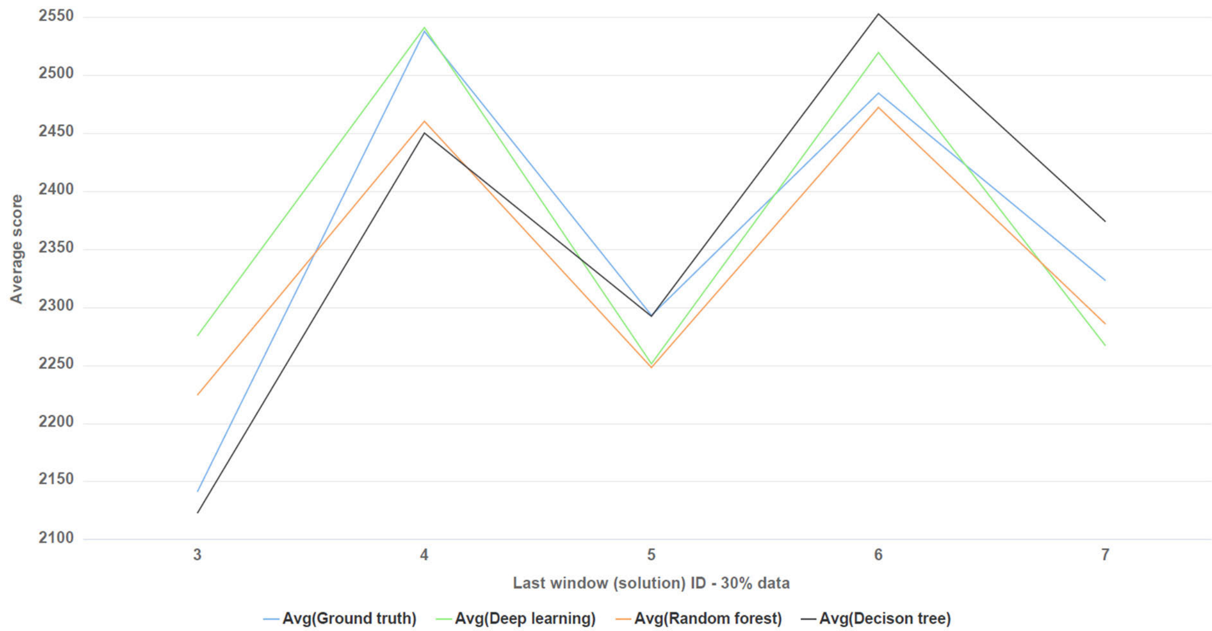
Source	F					P value				
	30% of solutions	*40% of solutions*	*50% of solutions*	*60% of solutions*	*70% of solutions*	*30% of solutions*	*40% of solutions*	*50% of solutions*	*60% of solutions*	*70% of solutions*
Between	9.028	12.718	13.282	15.743	16.689	.000*	.000*	.000*	.000*	.000*

*Differences between actual mean values are significant since the P value is smaller than Alpha = 0.05

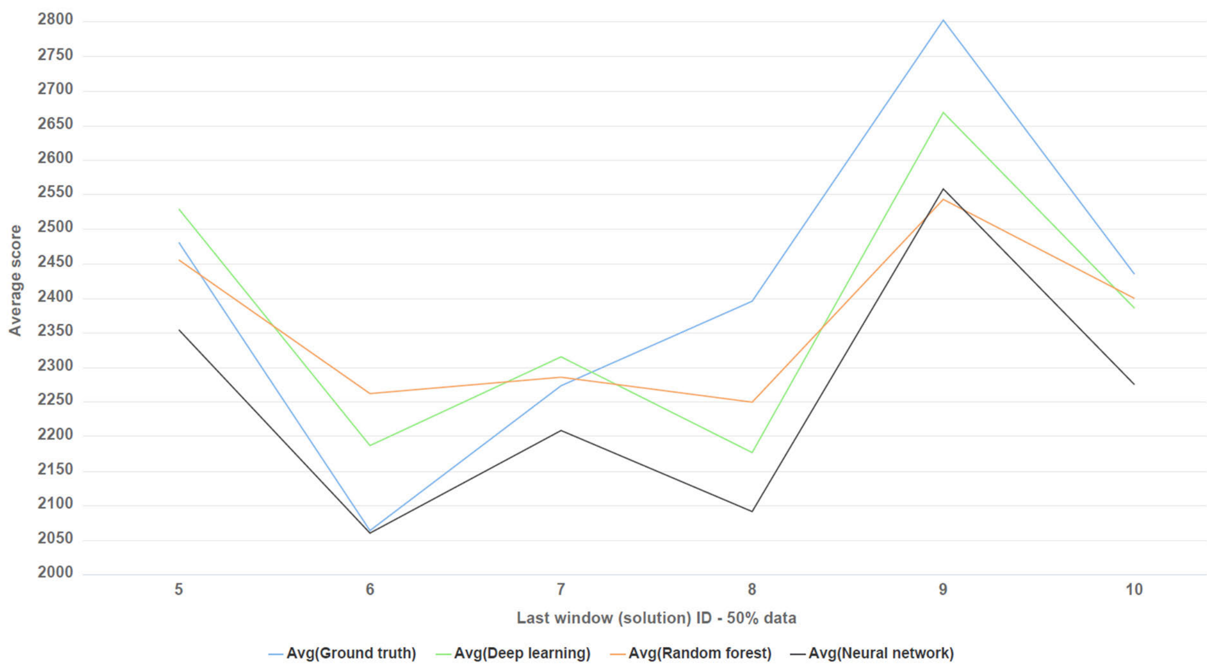
the measurement. To understand the RMSE values better, one should consider that the normalized RMSE can be calculated by dividing the reported RMSEs by the data range. Regarding R^2 values, it ranges from 0 to 1 and the closer the R^2 value to 1, the better a model fits a dataset. Combining all three metrics helps to take different aspects of a model into account, providing a robust basis for evaluating it. To further compare the performance vectors of the models, we performed statistical hypothesis testing using a pairwise t-test to determine the probability of the null hypothesis. The test was applied to all possible pairs of performance vectors, resulting in a significance matrix. One potential issue with implementing multiple paired t-tests is the increased chance of committing error type I. To relax this issue and determine whether the null hypothesis is wrong, we further applied an ANOVA test.

C. PERFORMANCE OF MODELS USING CROSS-VALIDATION: VALIDATION PHASE

Table 2 shows the performance of regression models (using three metrics of R^2, RMSE, and RE) for different sequence



(a)

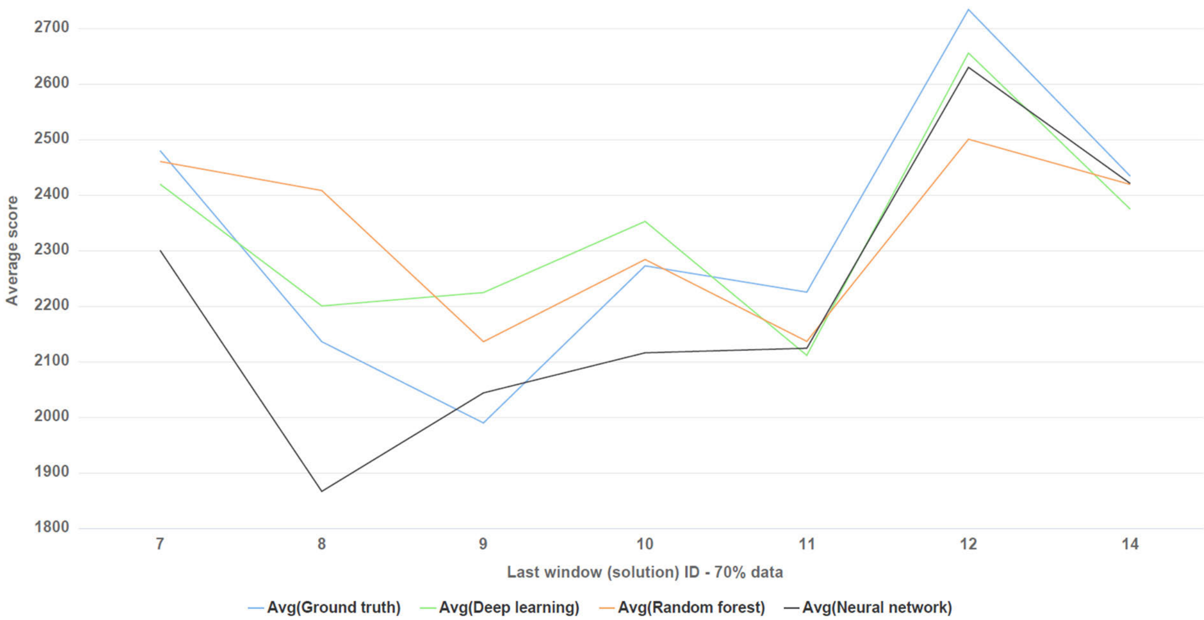


(b)

FIGURE 3. **Predictions using: a) 30%, b) 50%, and c) 70% of players' solutions from the top three models along with the ground truth using independent games data.**

lengths of actions using 10-fold cross-validation on data from 309 learners (including 5611 solutions). Additionally, as the focus of this research is on the early prediction of learners' performance during gameplay, Fig. 2 illustrates predictions using 30%, 40%, 50%, 60%, and 70% of learner solutions from the top three models along with the ground truth. Appendix A provides line charts showing the top three models for sequence lengths of 80%, 90%, and 100% of the learner's action (see Fig. 4).

Overall, it is apparent that the deep learning model outperforms other models in the early prediction of learner scores using various action sequence lengths, with a relative error below 10%. The smaller sequence lengths produced rather lower results and an increment in the sequence of



(c)

FIGURE 3. ***(Continued.)*** **Predictions using: a) 30%, b) 50%, and c) 70% of players' solutions from the top three models along with the ground truth using independent games data.**

the learners' actions resulted in better performance. The highest and lowest performance was achieved by the deep learning and support vector machines using 70% and 30% of solutions, respectively. Even though 100% of solutions included almost all the sequences of the learner's actions, they could not offer optimal predictive performance. Nonetheless, differences between the model performances are not substantial. Given the fact that our proposed approach aims to early predict learners' performance in educational games, the reported results in both Table 2 and Fig. 2 indicate that learners' actions during gameplay could be used to successfully predict their final score in a timely manner.

More specifically, for the prediction of learner's final score using 30% of their action sequence or solution submissions (e.g., 3 and 7 solutions for learners that completed the game with 10 and 20 solutions submission, respectively), the deep learning model showed a better performance with R^2 of 0.754, RMSE of 233.033, and RE of 9.27%. Likewise, using 40% and half of the learners' solution submissions, the deep learning model appeared to have the best performance with R^2 of 0.802 and 0.816, RMSE of 198.334 and 190.904, and RE of 7.67% and 7.45%, respectively. These results show that using less or 50% of learners' solutions, our proposed approach can predict learners' final scores with an R^2 of more than 0.75, indicating a very high correlation between predicted values and actual values. Moreover, training the deep learning model on 60% and 70% of learners' action sequences resulted in R^2 of 0.843 and 0.847, RMSE of 179.961 and 176.095, and RE of 6.96% and 6.62%, respectively. As shown in Fig. 2a to 2e, the random forest appeared to be the second-best model for the prediction of learners' final scores using 30% to 70% of their action sequences. Consequently, based on all three performance measures and their rather low standard deviation (computed from the 10-model accuracy), it can be said that the deep learning model can robustly predict the final score of learners when they complete 30% to 70% of their solutions. Our findings are consistent with those reported by Liu et al. [34], Hajra et al. [35], and Baranyi et al. [36] in that deep neural networks could outperform traditional machine learning techniques in prediction tasks of learner performance. While like the work reported by Chen and Cui [37] and Okubo et al. [38], we found that longer sequences of learner's actions could improve the prediction power of deep learning models, our results show that they may not necessarily be able to provide the optimal predictive performance. One possible reason which could be considered as a limitation of our approach is the use of multilayer perceptron rather than recurrent neural networks for handling sequential data. For processing sequential data, recurrent neural networks could be more suitable as they can capture sequential dependencies in the data. Increasing the sequences of actions could make models like multilayer perceptron and traditional machine learning more complex, and result in degrading their predictive performance. Additionally, our findings revealed that the prediction power of the deep learning model for learners with the lowest number of action sequences is as robust and accurate as those with the highest number of action sequences. However, this in many cases does not apply to other models, even those with the best performance after the deep learning model (e.g., random forest).

TABLE 5. Squared correlation, RMSE, and relative error of the cross-validated models for different sequence lengths using unseen data.

Models	Metrics	Sequence lengths		
		30% of solutions	*50% of solutions*	*70% of solutions*
Deep learning	R^2	0.861	0.891	0.872
	RMSE	176.934	176.882	181.915
	RE	5.84%	6.42%	6.88%
Neural network	R^2	0.610	0.846	0.860
	RMSE	299.554	249.817	213.499
	RE	11.71%	8.51%	7.33%
Linear regression	R^2	0.674	0.657	0.674
	RMSE	263.788	301.091	295.924
	RE	10.97%	12.41%	10.52%
KNN	R^2	0.710	0.781	0.734
	RMSE	260.807	293.563	309.876
	RE	9.87%	11.52%	12.33%
SVM	R^2	0.598	0.537	0.514
	RMSE	438.791	493.166	494.326
	RE	18.14%	19.61%	19.66%
Random forest	R^2	0.837	0.855	0.813
	RMSE	200.046	255.029	265.518
	RE	7.58%	10.02%	10.23%
Decision tree	R^2	0.742	0.600	0.449
	RMSE	260.685	321.362	390.948
	RE	8.93%	12.66%	13.31%
Generalized linear model	R^2	0.656	0.644	0.687
	RMSE	397.308	450.514	443.110
	RE	16.96%	17.98%	17.65%
Gradient boosted model	R^2	0.820	0.812	0.726
	RMSE	321.549	384.152	386.374
	RE	13.76%	15.30%	15.39%

* Squared correlation = R^2
**Relative error = RE

Table 3 reports the t-test results. Observe that the table only includes the test result for the deep learning model versus other models as it was found to be the best-performing model using all three performance measures. According to the results, except in the case of 30% of learners' solutions that a significant difference was not found between the deep learning model and the random forest, there is a significant difference between the deep learning model and all other models' mean values in all sequence lengths. Results of the follow-up ANOVA test, shown in Table 4, confirmed that differences between actual means are significant (rejecting the null hypothesis) providing evidence that the observed differences are likely due to differences in the models.

D. PERFORMANCE OF MODELS USING INDEPENDENT GAMES DATA: APPLICATION PHASE

This section reports our findings on applying the cross-validated models to an independent game dataset of 30 learners (including 560 solutions). To merely focus on the performance of models on the early prediction of learners' performance, we compared the performance of different models on three sequences of learner's action, which are 30%, 50%, and 70% of solutions (see Table 6 in Appendix A for model performances on other sequence lengths). Table 5 illustrates the performance of regression models using the three metrics.

The deep learning model appears to have outperformed other models in all three sequence lengths with an RE of less than 7%. More specifically, in sequence length of 30, the deep learning model reached R^2 of 0.861 and a relative error of 5.84%, with 176.934 RMSE. Adding 20% of solutions (i.e., 50% of solutions) improved the performance of the deep learning model even further where it could achieve R^2 as high as 0.891, RMSE of 176.882, and RE as low as 5.84%, respectively. Like the validation phase, the random forest model has consistently shown to be the second/third best

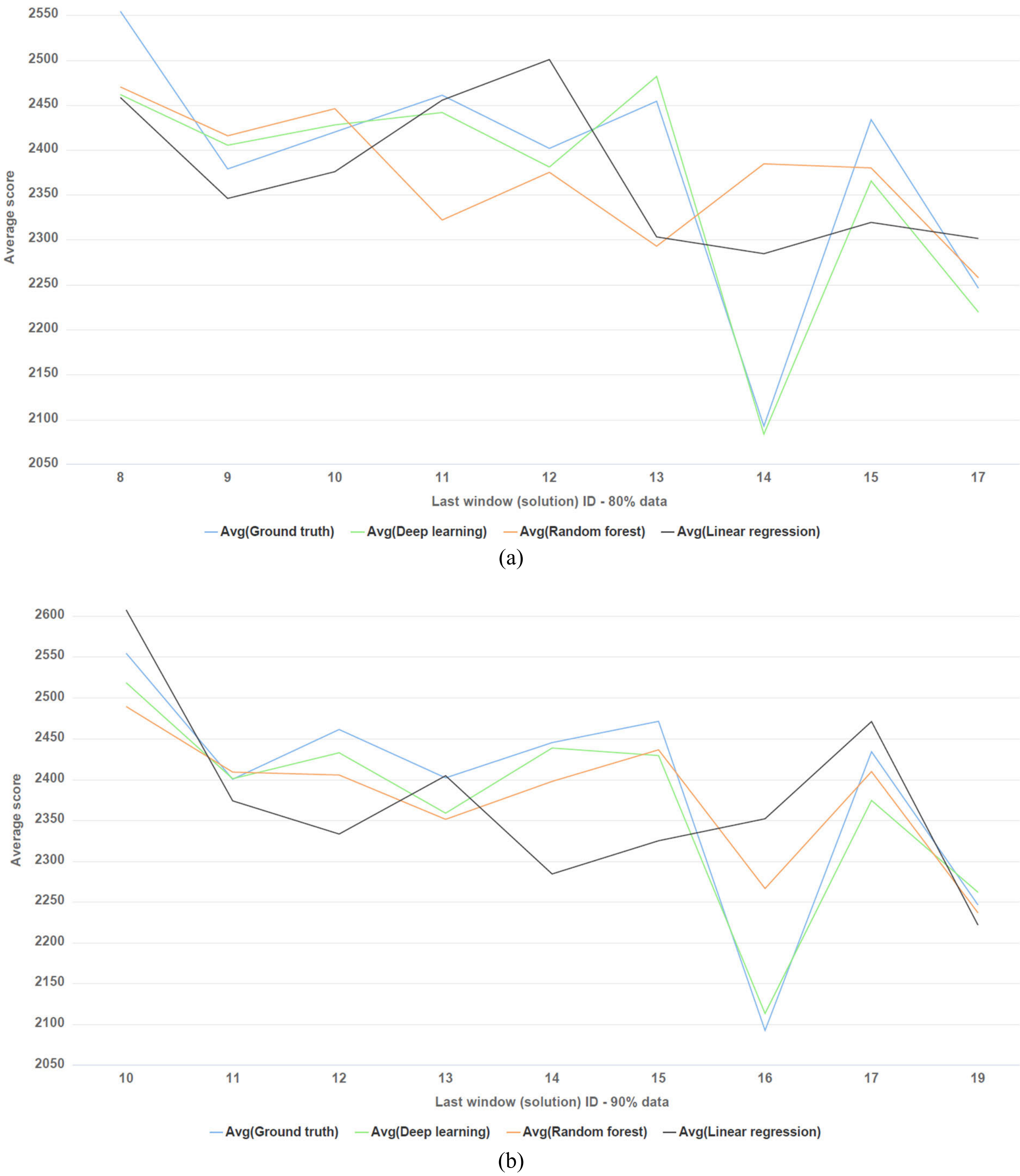
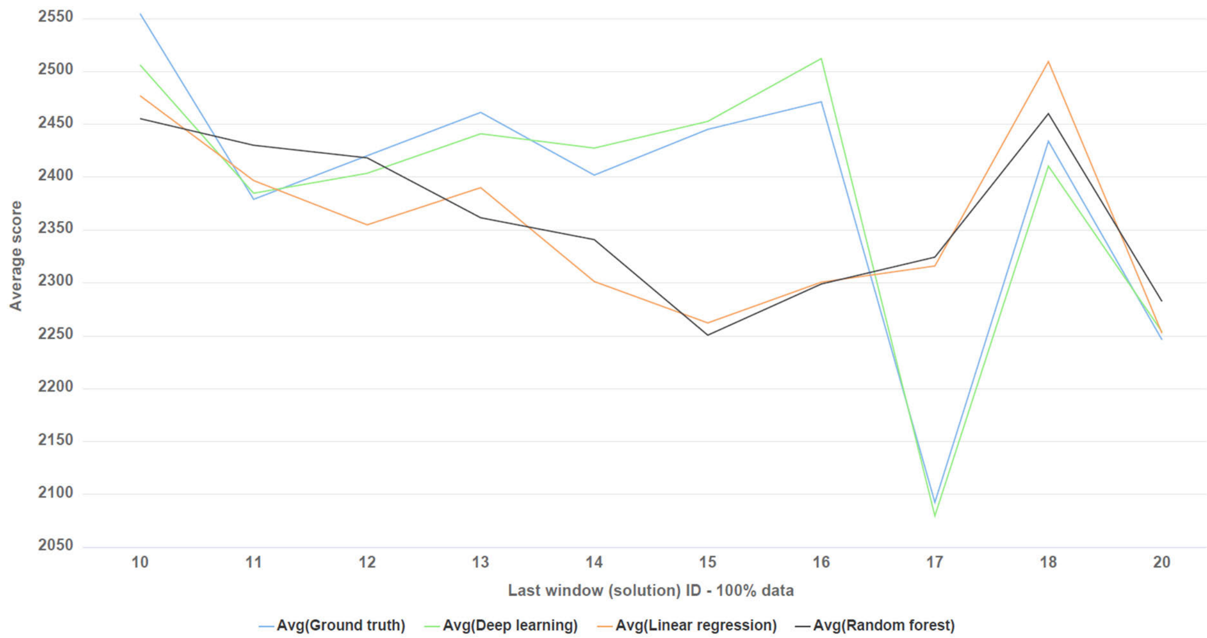


FIGURE 4. **Predictions using: a) 80%, b) 90%, and c) 100% of players' solutions from the top three models along with the ground truth using cross-validation.**

model. Fig. 3 illustrates predictions using 30%, 50%, and 70% of learners' solutions from the top three models along with the ground truth. Fig. 5 in Appendix A provides line charts showing the top three models for sequence lengths of 40%, 60%, 80%, 90%, and 100% of the learners' actions. The high performance of the validated models on the independent game dataset provides evidence for the applicability of the trained models (deep learning, and random forest in particular) to early predict learners' performance. Particularly, the model using only 30% of learners' action sequences was as competitive as those with more data, indicating its potential for early detection of at-risk learners.

These findings are in line with those reported by Gagnon et al. [11], Chen and Cui [37], Kim et al. [39],



(c)

FIGURE 4. ***(Continued.)*** **Predictions using: a) 80%, b) 90%, and c) 100% of players' solutions from the top three models along with the ground truth using cross-validation.**

and Okubo et al. [40] in that deep learning models could consistently achieve performance better than other methods during both validation and test stages while tackling classification task of early prediction of student performance. Unlike their work and many other related works (see Rastrollo-Guerrero et al. [41] for a recent review of predicting students' performance), our work also found that regardless of the lengths of learners' action sequences, the deep learning model could accurately and robustly handle the regression task of early prediction of learner performance at the early stages of the game.

V. CONCLUSION

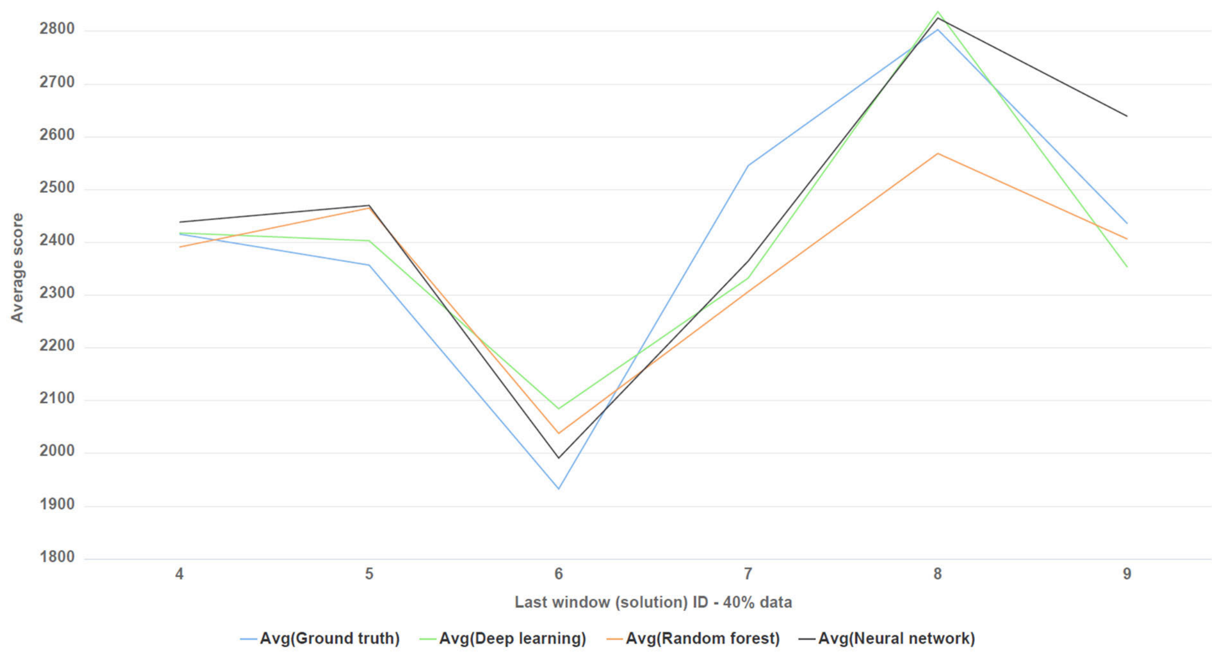
In this research, we proposed an approach, called GameEPM, for early prediction of learners' performance. In doing so, we examined the potential of a deep learning approach for the regression problem of early prediction of learners' final scores in the educational computer game of AutoThinking. Furthermore, considering the limited research comparing deep learning models with conventional machine learning methods for early prediction of learners' performance in educational computer games, we compared the performance of the deep learning model with eight machine learning methods from different families, including lazy, tree-based, neural network, function-based, and regression. Briefly, the GameEPM approach uses features related to learners' interactions with games and creates a series of time windows for a different sequence of actions to facilitate the identification of learners' performance (i.e., the final score in a game episode) in each time window. Once the time windows have been generated from a sequence of learners' actions, their score in each game episode is used to signal their final performance. To evaluate the feasibility of the GameEPM, we performed two sets of experiments using cross-validation and an independent game dataset. Regarding the first research question, our experiment results showed that the GameEPM approach could at early stages predict the learners' final score in the upcoming game episodes using the cross-validated deep learning model, with R^2 as high as 0.816 in the validation phase and 0.891 in the application phase. Concerning the second research question, a comparison of the results of different traditional machine learning methods with the deep learning model showed that, in the validation and application phases, up to 0.30 and 0.35 R^2 gain can be achieved by using the deep learning model, respectively. Regardless of the nature of the datasets (5611 solutions in the dataset used in the validation phase, and 560 solutions in the dataset used in the application phase) and the sequence of learners' actions, the deep learning model consistently outperforms all other models.

With respect to the third research question, using only 30% to 50% of learners' action sequences, our proposed approach could achieve performance as competitive as those with more data (i.e., $> 50\%$), indicating its potential for early detection of at-risk learners. For instance, we found that using only the first three solutions of learners who completed the game with 10 solutions, our approach can predict their final with R^2 around 0.861 and a relative error as low as 5.84%. Not only could the GameEPM accurately and stably predict learners' final scores during the validation phase, but also could slightly perform better during the application

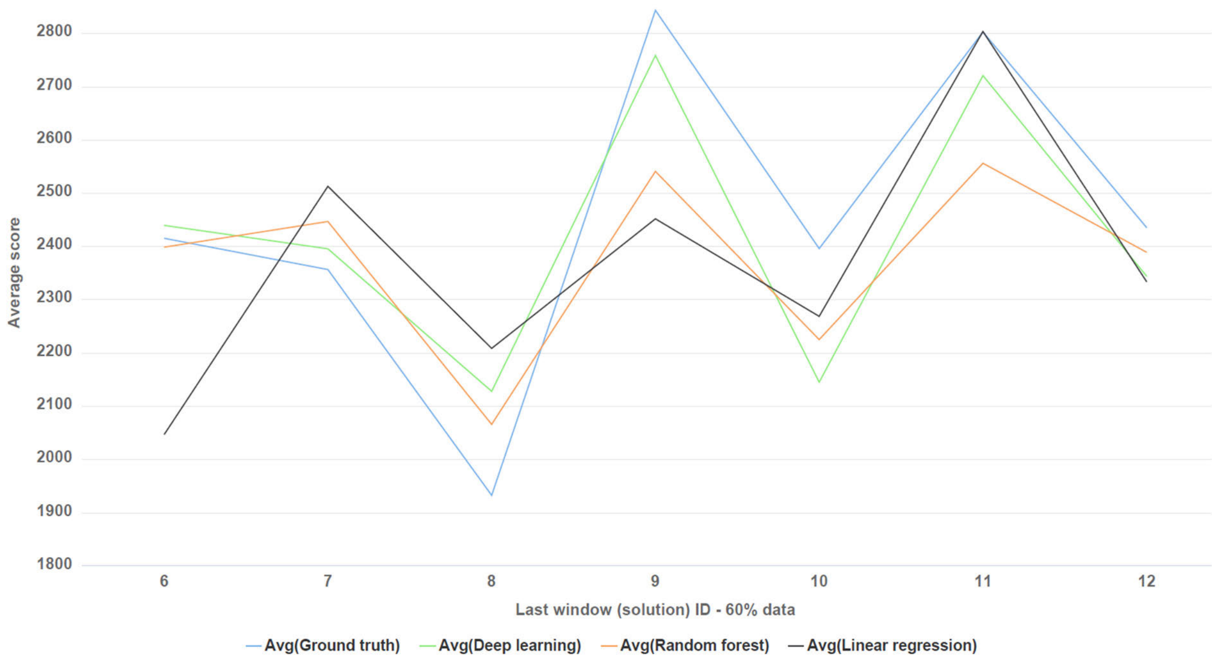
TABLE 6. Performance of model on other sequence lengths.

Models	Metrics	Sequence lengths				
		40% of solutions	*60% of solutions*	*80% of solutions*	*90% of solutions*	*100% of solutions*
Deep learning	R^2	0.874	0.870	0.869	0.876	0.878
	RMSE	183.703	186.904	187.838	184.600	177.761
	RE	6.69%	7.65%	7.23%	6.48%	6.48%
Neural network	R^2	0.788	0.773	0.792	0.863	0.798
	RMSE	238.161	303.962	263.048	230.064	266.897
	RE	8.79%	11.24%	10.34%	8.58%	9.05%
Linear regression	R^2	0.582	0.714	0.744	0.854	0.763
	RMSE	351.432	271.423	276.440	199.087	275.046
	RE	13.93%	11.04%	10.77%	6.99%	9.82%
KNN	R^2	0.782	0.724	0.727	0.801	0.765
	RMSE	284.382	307.584	304.043	281.422	285.873
	RE	11.11%	11.94%	12.00%	11.19%	11.86%
SVM	R^2	0.539	0.567	0.617	0.591	0.582
	RMSE	492.652	493.429	494.230	494.053	494.325
	RE	19.59%	19.60%	19.66%	19.66%	19.67%
Random forest	R^2	0.807	0.806	0.838	0.811	0.833
	RMSE	262.648	268.540	255.130	247.494	245.188
	RE	9.86%	10.10%	10.02%	9.23%	9.43%
Decision tree	R^2	0.481	0.692	0.615	0.663	0.745
	RMSE	364.214	280.713	319.663	295.077	258.630
	RE	12.85%	10.62%	12.84%	10.75%	9.37%
Generalized linear model	R^2	0.610	0.660	0.732	0.755	0.752
	RMSE	459.518	450.306	434.425	426.298	420.827
	RE	18.39%	17.94%	17.33%	16.99%	16.76%
Gradient boosted model	R^2	0.599	0.677	0.762	0.769	0.823
	RMSE	416.102	401.097	396.660	383.197	387.312
	RE	16.30%	15.85%	15.90%	14.98%	15.58%

phase. In some cases, the smaller sequence lengths produced slightly lower results and an increment in the sequence of learners' actions resulted in better performance. During the validation *phase*, the highest and lowest performance was achieved by the deep learning and support vector machines using 70% and 30% of solutions, while during the application phase, the deep learning and support vector machine showed the highest and lowest performance using 70% of solutions, respectively. This provides evidence for the potential of deep learning approaches in the regression problem of early



(a)



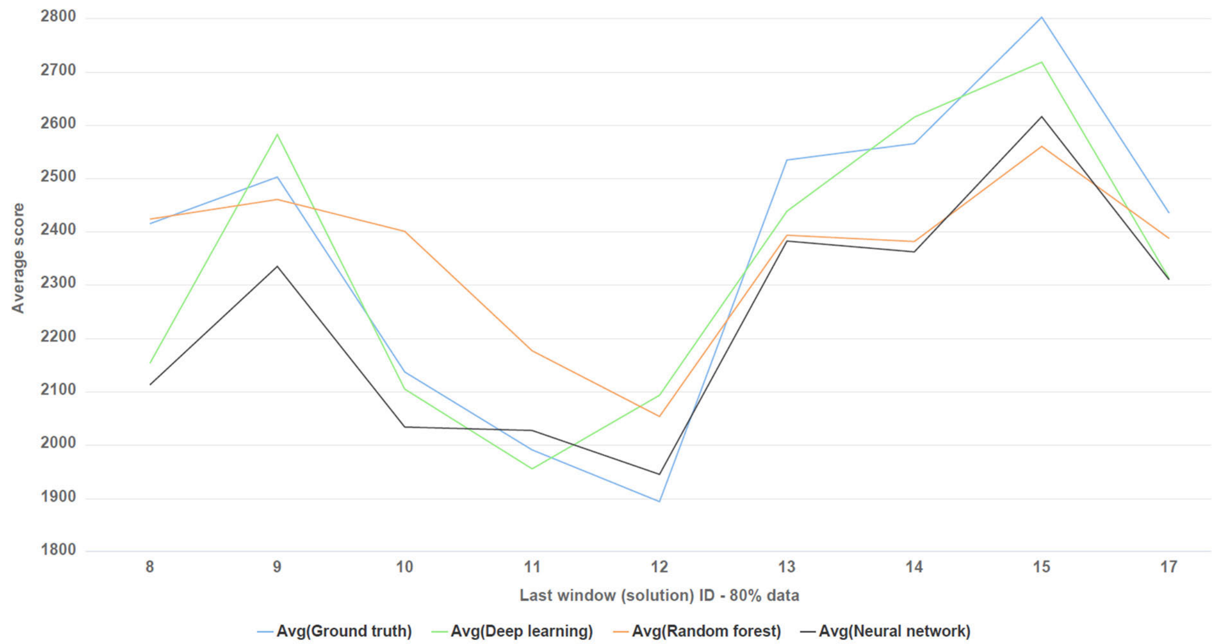
(b)

FIGURE 5. Predictions using: a) 40%, b) 60%, c) 80%, d) 90%, and e) 100% of players' solutions from the top three models along with the ground truth using cross-validation.

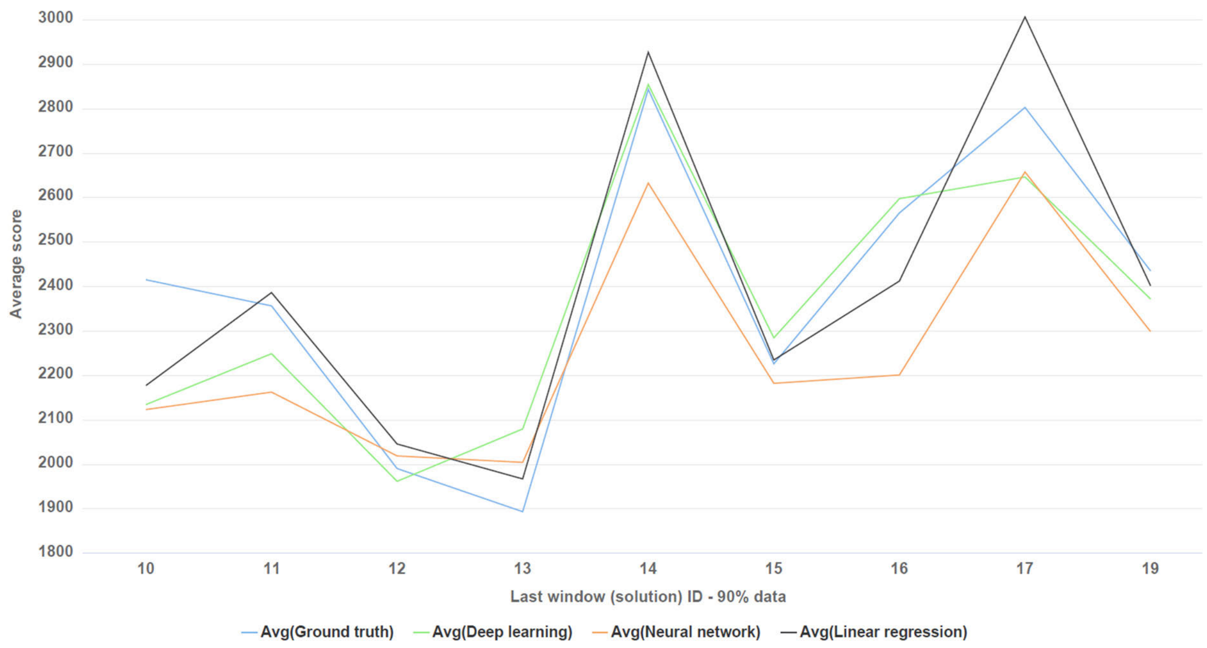
prediction of learners' performance in educational computer games.

Our results are in line with those reported by Gagnon et al. [11] and Lee-Cultura et al. [13] in that learners' data during gameplay can be used to accurately early predict their performance in educational games. However, the GameEEP approach performs even better than other extant approaches as it benefits from a deep learning-based method that effectively deals with modelling latent information of learners according to their in-game data. Moreover, the effectiveness of our proposed approach is shown in two phases, validation and application. Needless to mention that the GameEPM tackles the regression problem of early prediction of learners' final score rather than the binary classification task.

Given the main goal of learning/predictive analytics which is early prediction, we believe that using the entire learner



(c)

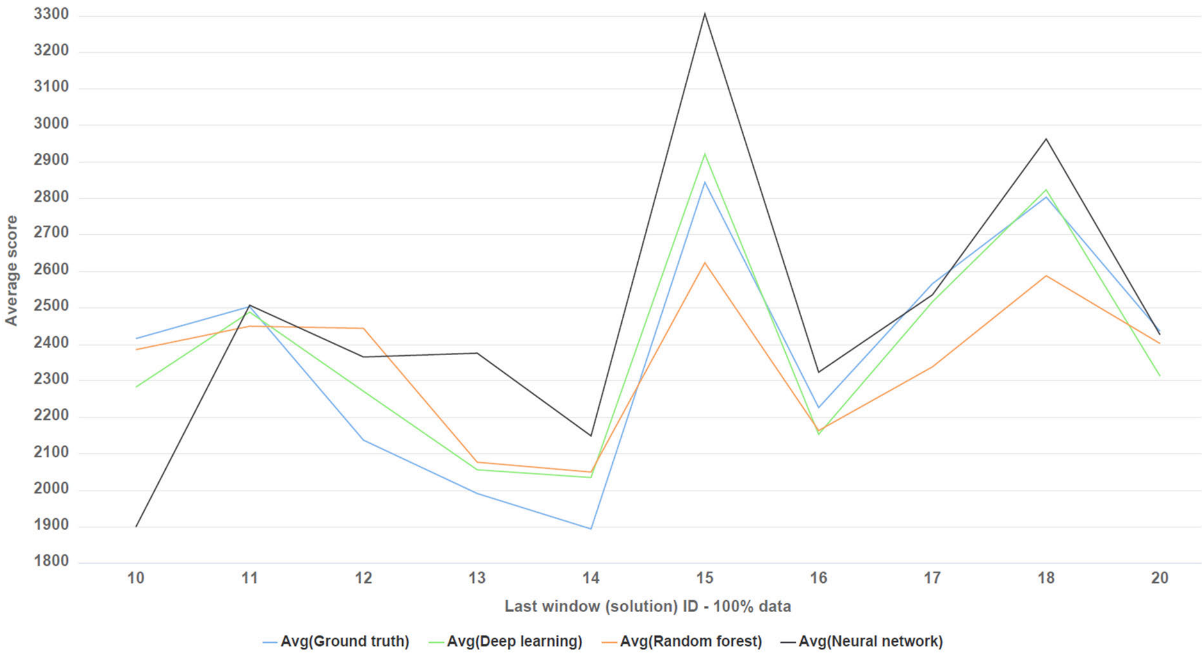


(d)

FIGURE 5. ***(Continued.)*** **Predictions using: a) 40%, b) 60%, c) 80%, d) 90%, and e) 100% of players' solutions from the top three models along with the ground truth using cross-validation.**

data neither is practical nor helpful to improving the predictive power of predictive models for early prediction of learners' performance in educational games. Considering the high prediction power of our proposed approach, future works could consider: i) addressing the early prediction of learner performances as a regression problem; ii) exploiting the strength of deep neural networks in encoding learners' knowledge and behaviour and accordingly predicting their performance at the early stages of educational games; iii) investigating the effect of various sequences of learner's actions on performance of early predictive modelling and employing the most appropriate for the task in hand.

Some limitations of our research could also be regarded as directions for future research. For instance, we merely



(e)

FIGURE 5. ***(Continued.)*** **Predictions using: a) 40%, b) 60%, c) 80%, d) 90%, and e) 100% of players' solutions from the top three models along with the ground truth using cross-validation.**

validated the performance of our proposed approach using data from a single educational game. To investigate the generalizability of our approach, it would be useful to apply the GameEPM approach to data from other games. Additionally, future works could consider improving the accuracy of the GameEPM approach by including side information on learners. Finally, it would be useful to integrate the proposed approach into the AutoThinking game in order to provide optimal learning experiences and accordingly evaluate its effectiveness in real-world settings.

APPENDIX

See Figs. 4, 5, and Table 6.

REFERENCES

[1] N. El Mawas, D. Hooshyar, and Y. Yang, "Effects of adaptive educational games on adults' computational thinking," in *Proc. Int. Conf. Comput. Supported Educ.* Cham, Switzerland: Springer, 2020, pp. 162–176.

[2] K. L. Whitney, "The effect of personalized learning on student achievement," Ph.D. thesis, Regent Univ., Virginia Beach, VA, USA, 2021.

[3] L. Bennis, K. Kandali, and H. Bennis, "Studying learner's player learning style for generating adaptive learning game," *IEEE Access*, vol. 10, pp. 103880–103887, 2022.

[4] A. Tlili, M. Denden, F. Essalmi, M. Jemni, N.-S. Chen, and R. Huang, "Does providing a personalized educational game based on personality matter? A case study," *IEEE Access*, vol. 7, pp. 119566–119575, 2019.

[5] D. Hooshyar, M. Pedaste, Y. Yang, L. Malva, G.-J. Hwang, M. Wang, H. Lim, and D. Delev, "From gaming to computational thinking: An adaptive educational computer game-based learning approach," *J. Educ. Comput. Res.*, vol. 59, no. 3, pp. 383–409, Jun. 2021.

[6] R. Ghali, C. Frasson, and S. Ouellet, "Towards real time detection of learners' need of help in serious games," in *Proc. 29th Int. Flairs Conf.*, 2016, pp. 1–4.

[7] D. Hooshyar, M. Yousefi, and H. Lim, "Data-driven approaches to game player modeling: A systematic literature review," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–19, 2018.

[8] D. Hooshyar, Y.-M. Huang, and Y. Yang, "GameDKT: Deep knowledge tracing in educational games," *Expert Syst. Appl.*, vol. 196, Jun. 2022, Art. no. 116670.

[9] A. Hellas, P. Ihantola, A. Petersen, V. V. Ajanovski, M. Gutica, T. Hynninen, A. Knutas, J. Leinonen, C. Messom, and S. N. Liao, "Predicting academic performance: A systematic literature review," in *Proc. Companion 23rd Annu. ACM Conf. Innov. Technol. Comput. Sci. Educ.*, Jul. 2018, pp. 175–199.

[10] D. Hooshyar, M. Pedaste, and Y. Yang, "Mining educational data to predict students' performance through procrastination behavior," *Entropy*, vol. 22, no. 1, p. 12, 2019.

[11] D. J. Gagnon, E. Harpstead, and S. Slater, "Comparison of off the shelf data mining methodologies in educational game analytics," in *Proc. EDM (Workshops)*, 2019, pp. 38–43.

[12] L. P. Macfadyen and S. Dawson, "Mining LMS data to develop an 'early warning system' for educators: A proof of concept," *Comput. Educ.*, vol. 54, no. 2, pp. 588–599, 2010.

[13] S. Lee-Cultura, K. Sharma, S. Papavlasopoulou, and M. Giannakos, "Motion-based educational games: Using multi-modal data to predict player's performance," in *Proc. IEEE Conf. Games (CoG)*, Aug. 2020, pp. 17–24.

[14] J. Wiggins, M. Kulkarni, W. Min, B. Mott, and K. Boyer, "Affect-based early prediction of player mental demand and engagement for educational games," in *Proc. AAAI Conf. Artif. Intell. Interact. Digit. Entertainment*, 2018, vol. 14, no. 1, pp. 243–249.

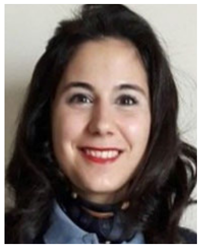
[15] M. Milošević, N. Živić, and I. Andjelković, "Early churn prediction with personalized targeting in mobile social games," *Expert Syst. Appl.*, vol. 83, pp. 326–332, Oct. 2017.

[16] M. Bienkowski, M. Feng, and B. Means, "Enhancing teaching and learning through educational data mining and learning analytics: An issue brief," Office Educ. Technol., U.S. Dept. Educ., Washington, DC, USA, Tech. Rep., Oct. 2012. Accessed: Oct. 13, 2022. [Online]. Available: https://eric.ed.gov/?id=ED611199

[17] V. Shute and M. Ventura, *Stealth Assessment: Measuring and Supporting Learning in Video Games*. Cambridge, MA, USA: MIT Press, 2013.

[18] M. Tan and P. Shao, ''Prediction of student dropout in e-learning program through the use of machine learning method,'' *Int. J. Emerg. Technol. Learn.*, vol. 10, no. 1, pp. 1–7, 2015.
[19] A. P. Patil, K. Ganesan, and A. Kanavalli, ''Effective deep learning model to predict Student grade point averages,'' in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res. (ICCIC)*, Dec. 2017, pp. 1–6.
[20] Z. Iqbal, J. Qadir, A. Noor Mian, and F. Kamiran, ''Machine learning based Student grade prediction: A case study,'' 2017, *arXiv:1708.08744*.
[21] S. D. A. Bujang, A. Selamat, R. Ibrahim, O. Krejcar, E. Herrera-Viedma, H. Fujita, and N. A. M. Ghani, ''Multiclass prediction model for student grade prediction using machine learning,'' *IEEE Access*, vol. 9, pp. 95608–95621, 2021.
[22] D. Liu, Y. Zhang, J. Zhang, Q. Li, C. Zhang, and Y. Yin, ''Multiple features fusion attention mechanism enhanced deep knowledge tracing for Student performance prediction,'' *IEEE Access*, vol. 8, pp. 194894–194903, 2020.
[23] R. Alamri and B. Alharbi, ''Explainable student performance prediction models: A systematic review,'' *IEEE Access*, vol. 9, pp. 33132–33143, 2021.
[24] M. Tadayon and G. J. Pottie, ''Predicting student performance in an educational game using a hidden Markov model,'' *IEEE Trans. Educ.*, vol. 63, no. 4, pp. 299–304, Nov. 2020.
[25] C. Alonso-Fernández, I. Martínez-Ortiz, R. Caballero, M. Freire, and B. Fernández-Manjón, ''Predicting students' knowledge after playing a serious game based on learning analytics data: A case study,'' *J. Comput. Assist. Learn.*, vol. 36, no. 3, pp. 350–358, Jun. 2020.
[26] J. Tang and N. R. Prestopnik, ''Science gamers, citizen scientists, and dabblers: Characterizing player engagement in two citizen science games,'' *Int. J. Hum.-Comput. Interact.*, to be published.
[27] M. D. Kickmeier-Rust, E. Mattheiss, C. Steiner, and D. Albert, ''A psycho-pedagogical framework for multi-adaptive educational games,'' *Int. J. Game-Based Learn.*, vol. 1, no. 1, pp. 45–58, Jan. 2011.
[28] J. L. Plass, B. D. Homer, S. Pawar, C. Brenner, and A. P. MacNamara, ''The effect of adaptive difficulty adjustment on the effectiveness of a game to develop executive function skills for learners of different ages,'' *Cognit. Develop.*, vol. 49, pp. 56–67, Jan./Mar. 2019.
[29] S. Sampayo-Vargas, C. J. Cope, Z. He, and G. J. Byrne, ''The effectiveness of adaptive difficulty adjustments on students' motivation and learning in an educational computer game,'' *Comput. Educ.*, vol. 69, pp. 452–462, Nov. 2013.
[30] C. Roberto, Y. Shi, A. Amanmyradova, and S. Meghani, ''Predicting student's post-test performance using game data,'' Tech. Rep.
[31] D. Hooshyar and Y. Yang, ''Predicting course grade through comprehensive modelling of students' learning behavioral pattern,'' *Complexity*, vol. 2021, May 2021, Art. no. 7463631.
[32] N. El Mawas, D. Hooshyar, and Y. Yang, ''Investigating the learning impact of autothinking educational game on adults: A case study of France,'' in *Proc. CSEDU*, 2020, pp. 188–196.
[33] D. Hooshyar, H. Lim, M. Pedaste, K. Yang, M. Fathi, and Y. Yang, ''AutoThinking: An adaptive computational thinking game,'' in *Proc. Int. Conf. Innov. Technol. Learn.*, 2019, pp. 381–391.
[34] Y. Liu, S. Fan, S. Xu, A. Sajjanhar, S. Yeom, and Y. Wei, ''Predicting Student performance using clickstream data and machine learning,'' *Educ. Sci.*, vol. 13, no. 1, p. 17, Dec. 2022.
[35] H. Waheed, S.-U. Hassan, N. R. Aljohani, J. Hardman, S. Alelyani, and R. Nawaz, ''Predicting academic performance of students from VLE big data using deep learning models,'' *Comput. Hum. Behav.*, vol. 104, Mar. 2020, Art. no. 106189.
[36] M. Baranyi, M. Nagy, and R. Molontay, ''Interpretable deep learning for university dropout prediction,'' in *Proc. 21st Annu. Conf. Inf. Technol. Educ.*, Oct. 2020, pp. 13–19.
[37] F. Chen and Y. Cui, ''Utilizing student time series behaviour in learning management systems for early prediction of course performance,'' *J. Learn. Anal.*, vol. 7, no. 2, pp. 1–17, 2020.
[38] F. Okubo, T. Yamashita, A. Shimada, and H. Ogata, ''A neural network approach for students' performance prediction,'' in *Proc. 7th Int. Learn. Anal. Knowl. Conf.*, Mar. 2017, pp. 598–599.
[39] B.-H. Kim, E. Vizitei, and V. Ganapathi, ''GritNet: Student performance prediction with deep learning,'' 2018, *arXiv:1804.07405*.
[40] F. Okubo, T. Yamashita, A. Shimada, and S. Konomi, ''Students' performance prediction using data of multiple courses by recurrent neural network,'' in *Proc. 25th Int. Conf. Comput. Educ. (ICCE)*, 2017, pp. 439–444.
[41] J. L. Rastrollo-Guerrero, J. A. Gómez-Pulido, and A. Durán-Domínguez, ''Analyzing and predicting students' performance by means of machine learning: A review,'' *Appl. Sci.*, vol. 10, no. 3, p. 1042, 2020.

DANIAL HOOSHYAR received the Ph.D. degree from the Artificial Intelligence Department, University of Malaya, Malaysia, in 2016. He was a Research Professor with the Department of Computer Science and Engineering, Korea University, for nearly four years. He is currently an Associate Professor of learning analytics and educational data mining with Tallinn University, Estonia. His research interests include artificial intelligence in education, adaptive educational systems, and technology-enhanced learning and teaching.



NOUR EL MAWAS received the Ph.D. degree in computer science from UTT, Troyes, in 2013. She did her Ph.D. research in the design of instructional scenarios on learning management system (LMS) with the Université du Maine. She was a Researcher with Telecom Bretagne, where she worked on MOOCs personalization in a lifelong learning perspective. She was also a Researcher with the National College of Ireland (NCI), where she worked on innovative pedagogical methods that are used to teach STEM subjects. Currently, she is an Associate Professor with the Université de Lille, where her work focused on the design of serious games for expertise training in complex situations, such as crisis and sustainable development. Her research interests include technology-enhanced learning, learning personalization, serious games, learning management systems, MOOCs, and lifelong learning.

MARCELO MILRAD (Member, IEEE) is a full Professor of media technology with Linnaeus University (LNU), Sweden. Since March 2020, he has been acting as the main Scientific Coordinator of LNU's knowledge environment ''Digital Transformation.'' He is also actively involved in a new initiative, called ''Digital Humanities,'' with LNU along with his colleagues at the Faculty of Arts and Humanities. He conducts his research in very close collaboration with industrial partners and the public sector. He has published over 240 articles in international journals, refereed conferences, books, and technical reports. He has also been presenting and giving lectures about his work in more than 45 countries worldwide. His current research interests include technology-enhanced learning (TEL), advanced human–computer interaction, novel uses of big data techniques, and mobile technologies in education and healthcare.

YEONGWOOK YANG received the master's degree in computer science education and the Ph.D. degree from the Department of Computer Science and Engineering, Korea University, Seoul, South Korea. He was a Research Professor with the Department of Computer Science and Engineering, Korea University, for one year. He was a Senior Researcher with the University of Tartu, Tartu, Estonia. He is currently an Assistant Professor with the Division of Computer Engineering, Hanshin University. His research interests include information filtering, recommendation systems, educational data mining, and deep learning.