Courses Recommendation Algorithm Based On Performance Prediction In E-Learning

Dagou Dangui Augustin Sylvain Legrand Koffi [1†], Nouho Ouattara [2††], Digrais Moïse Mambe [3†††], Souleymane Oumtanaga [4††††], and Assohoun ADJE [5†††††]

[1†] and [5†††††] Unité de Formation et de Recherche des Mathématiques et Informatique (UFR-MI), Université Félix Houphouët-Boigny (UFHB), Abidjan, Côte d'Ivoire
[2††], [3†††] and [4††††] Laboratoire de Recherche en Informatique et Télécommunication (LARIT)
[2††] Université Félix Alassane Ouattara (UAO), Bouaké, Côte d'Ivoire
[3†††] Université Nangui Abrogoua (UNA), Abidjan, Côte d'Ivoire
[4††††] Institut National Polytechnique Félix Houphouët-Boigny (INPHB), Yamoussoukro, Côte d'Ivoire

Summary

The effectiveness of recommendation systems depends on the performance of the algorithms with which these systems are designed. The quality of the algorithms themselves depends on the quality of the strategies with which they were designed. These strategies differ from author to author. Thus, designing a good recommendation system means implementing the good strategies. It's in this context that several research works have been proposed on various strategies applied to algorithms to meet the needs of recommendations. Researchers are trying indefinitely to address this objective of seeking the qualities of recommendation algorithms. In this paper, we propose a new algorithm for recommending learning items. Learner performance predictions and collaborative recommendation methods are used as strategies for this algorithm. The proposed performance prediction model is based on convolutional neural networks (CNN). The results of the performance predictions are used by the proposed recommendation algorithm. The results of the predictions obtained show the efficiency of Deep Learning compared to the k-nearest neighbor (k-NN) algorithm. The proposed recommendation algorithm improves the recommendations of the learners' learning items. This algorithm also has the particularity of dissuading learning items in the learner's profile that are deemed inadequate for his or her training.

Key words:
Algorithm, deep learning, recommendation system, collaborative approach, e-learning

1. Introduction

Recommendation systems are computer applications whose role is to assist users in making decisions that meet their needs [1]. For their great importance, they are spread in several areas. The e-learning field uses recommendation systems to tailor training to the learner's profile. However, with the growth of e-learning, the amount of data is becoming increasingly important. This leads to difficulties in finding suitable learning objects for the learners [2]. Developers and researchers have been working for several years in the quest for solutions to improve the quality of recommendation systems [3] [4]. In order to achieve the set objectives, the proposal of good quality recommendation algorithms remains at the heart of the research [5] [6].

From this observation, one concern remains: What strategies should be put in place in order to provide algorithms for recommendation systems capable of reducing by a large margin the shortcomings in the recommendations?

In order to provide an answer to this question asked, an algorithm for recommending learning items is proposed in this work. This algorithm has two contributions. In the first contribution, it's a process that recommends the appropriate learning items for the learner's training. In the second contribution, it is about a process which advises against the inadequate learning items found in a learner's profile. To be efficient, this algorithm is based on the results of learner performance predictions. These predictions are based on convolutional neural networks (CNN).

The results obtained from the experiments in this work present the predictive performance of Deep Learning above that of the k-NN algorithm. Thus, the recommendations are improved in both contributions.
The structure of this paper is presented as follows: Related work is presented first in Section 2. Then the learning algorithms used in this article are presented in section 3. Section 4 presents our proposed recommendation algorithm. Section 5 presents an evaluation of our algorithm followed by a discussion. Section 6 concludes this manuscript.

Manuscript received February 5, 2021
Manuscript revised February 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.2.17

2. Related Work

Numerous studies have been carried out to improve the performance of learning resource recommendation systems. Thus, in 2013, in order to propose adequate courses to students for their training through recommendation systems, Aher and Lobo [7] show the usefulness of data mining techniques in these systems. A combination of clustering and association rule algorithm is used for this purpose. This system then allows recommendations to be made to a new user who recently signed up for a course. Also, the results correspond to the real interdependencies between the courses. One year later, in 2014, Aher [8] proposes a course recommendation system based on another combination. That of the algorithm for maximum grouping of pending elements (Expectation Maximization Clustering) and the association rule algorithm (Association Rule Algorithm). In 2015, Diem [9] uses logistic regression to propose a system for recommending appropriate resources by classification. His work responds to the problem of missing values. Thus, his system could be advised to choose certain adapted courses during a next semester.

In the same vein of recommendation, in 2016, two teams of researchers each propose an approach. First, Al-Badarenah and Alsakran [10] proposes a collaborative recommendation system based on a knowledge extraction algorithm. Then, Imran and his collaborators [11] work to personalize the recommendations according to the learners' profiles. Their proposed system uses the extraction of association rules. They claim that these kinds of systems can increase learner performance and satisfaction.
The following year, research continued, again on issues of data overload and the large user numbers in massive open online courses (MOOCs), which made it difficult to choose the appropriate learning resources. Thus, Xiao and his collaborators [12] propose a personalized learning object recommendation system based on a combinatorial algorithm to meet users' needs. In the same year, Shu and friends [13] used convolution neural network technology and text information to propose a model for recommending multimedia learning items. The aim of this model is to enable students to find new resources for their learning. The results of their work show improvements over conventional methods. Also, Liu [14] in front of traditional recommendation systems that suffer from poor quality recommendations through low scalability and lack of stability, proposes a solution. That of a new recommendation algorithm based on the theory of sets of influence. The proposed method generates a more accurate recommendation than traditional recommendation algorithms.

In 2018, Bourkoukou and Achbarou [15] are conducting a study to address the problems of cold start and data scarcity in e-learning systems. Their model would improve the quality of the recommendations, with useful content and minimum processing time. Their proposed method is based on the weighting of learning resources. Bourkoukou and Bachari [16], think that it is easy to recommend a set of learning objects from knowledge of the learner's profile. Thus, they offer a model that automatically adapts to the dynamic preferences of learners. First their model addresses the cold start problem by using the Felder and Silverman model. Then, through information on learners' interactions and actions, it analyzes learners' preferences and habits. Finally, he reviews and updates the learning senario via a hybrid recommendation system based on the extraction of association rules and K-Nearest Neighbors algorithms. The results of their work increase the quality of learning and satisfy the learner. Asadi and al. [17], in view of the students' difficulties in finding information on each course, have carried out work with the aim of proposing a model of course recommendations as a solution. They do this by taking into account the characteristics of the students by using clustering algorithms to determine similar students. Then, they use fuzzy association rules to determine dependencies between learners' course choices. Their study facilitates decision making in the choice of courses. Limitations in their work are, first of all, that students are forced to follow a particular order of learning because of the programming of core subjects. Then with the new courses inserted in the system cannot be personalized to a student for a recommendation. That same year, with the increase in learning resources on MOOC platforms, and the variation in cognitive capacity and knowledge structures, identifying resources of interest to learners is difficult. Thus, Zhang et al. [18] proposed a recommendation model based on deep belief networks in MOOC environments. The experimental results show that compared to traditional recommendation methods, the proposed model has greater precision. Pereira et al, [19] argue that it is problematic to make the relevant choice among educational resources when they are distributed among several repositories in a system. They thus propose a model based on information extraction techniques and semantic web techniques to first extract from the Facebook social network, the profile and the educational context of the users. Second, make resource recommendations. The results of the assessments show users' satisfaction and approval with the information extracted from them. Also, they allow to set up a network of interest between users on specific topics. Gulzar and al. [20] propose a model of a hybrid recommendation system with an ontology in order to collect useful information to make precise recommendations. Cette étude vise à trouver une solution au problème de manque de performances liées aux recommandations des systèmes de recommandation traditionnels.

In 2019, Wan and Niu [21] carried out work to find a solution to the problems of interpersonal information gaps between learners in recommender systems. This problem does not facilitate the application of collaborative filtering techniques for personalized and effective recommendations. Thus, they propose a hybrid filtering approach using a combination of recommendation strategy and sequential pattern mining (SPM) and learner influence model (LIM), based on self-organization based (SOB). In addition, they used fuzzy logic to optimize the learners' influence model. The results of their work made it possible to have good, diversified and personalized performance recommendations. Nafea and al. [22] are also thinking about finding a solution to improve learning object recommendations based on student preferences. Thus, they propose a new recommendation algorithm using the hybrid model of Felder and Silverman (Felder-Silverman Learning Styles Model (FSLSM)) by combining the actual notations and the learning styles of the students. The results of their experiments provide an improvement to the recommendations of e-learning systems to learners. Given the growth of e-learning which causes difficulties in finding adequate learning objects, Nafea and its collaborators [2] have been working to improve the customization of learning object recommendation systems by proposing an algorithm based on Felder and Silverman's learning model. This algorithm highlights students' learning styles and learning object profiles. Thus, they argue that the K-means algorithm, the Pearson correlation and the Cosine similarity measure are good tools for the implementation of recommendation systems. They plan to extend their algorithm in their future work to improve the accuracy of recommendations by taking into account the assessments of learners who have a learning style similar to that of the active learner.

Although other approaches are proposed in the literature, it should be remembered that systems for recommending learning resources should be reviewed regularly and create new models for recommendations.

3. Learning Algorithms

3.1 Deep Learning

Machine learning has been proven to work for several years. However, for the last decade, it has required a deepening in its operation because of its limitations in the processing of data in a raw form, in the face of the demands of complex system studies. Thus, meticulous research studies by experts have led to the implementation of Deep Learning [23].

Machine learning has several algorithms in its operation. One of the sets of these algorithms allows for deep learning. These are usually algorithms based on artificial neural networks. This deep learning takes place in several different layers, where the upper layers depend on the lower layers [24]. It should be noted that each level has a different setting or concept from the others. Deep Learning is a field that started in 2006. Its goal is to give meaning to data such as texts, images and audio in order to make Artificial Intelligence more applicable through supervised or unsupervised machine learning methods [25].

Complex non-linear functions are exploited today by deep learning because of its high signal processing performance and large amounts of data.

Work in the field of deep learning falls into three broad categories. They can be in the category of supervised learning, unsupervised learning or blended learning. The most widely used category for deep or non-deep learning is supervised learning [23].

3.2 k-NN Algorithm

The k-NN algorithm is a classical Machine Learning algorithm for regression [26]. It is a supervised algorithm that stores input information to provide an output [27]. It operates in two phases. First a training phase and then a testing phase. Moreover, for its implementation, it is necessary to take into account the determination of the distances to determine "k" nearest neighbors, k being an integer [28]. The determined distance is taken between the vectors of each user in our present case. Thus, the Minkowski distance between the vectors $X_1 = (x_{11}, x_{12}, \dots, x_{1n})$ and $X_2 = (x_{21}, x_{22}, \dots, x_{2n})$ is defined by Eq.(1) [29]:

$$dist(X_1, X_2) = \left(\sum_{i=1}^{n} \left| x_i - x_j \right|^p \right)^{1/p} \quad (1)$$

The number "k" for the nearest k neighbors is taken in to calculate the output as shown in Eq.(2) [28]:

$$Ouput = \frac{1}{k} \sum_{i=1}^{k} d_{pi} \quad (2)$$

With d_{pi} being the output for sample i.

The advantage of the k-NN algorithm is that it is stable and has good precision. However, it takes time in its execution because it has to calculate several distances according to the samples in the dataset, before determining the next neighbors [30].

4. Recommendation Process

This approach to recommending learning resources is done in two sub-processes. Each of the two sub-processes is based on a specific proposed algorithm. The first phase consists of determining and adding suitable learning items to the profile of a selected learner. The second process, unlike the first, allows learning items to be removed from the user's profile.
The proposed recommendation process consists of a number of tasks. First, it takes as input a matrix containing the learners and their predicted performance for each learning resource. Then, for a learner selected in the matrix, the algorithm selects each learning item of his profile in turn. The selected items go to the sub-process "recommend learning item" or to the sub-process "advise against learning item" respectively, if the learning item has not already been used by the learner or if the learning item has already been used by the learner. This process is shown in Fig. 1.

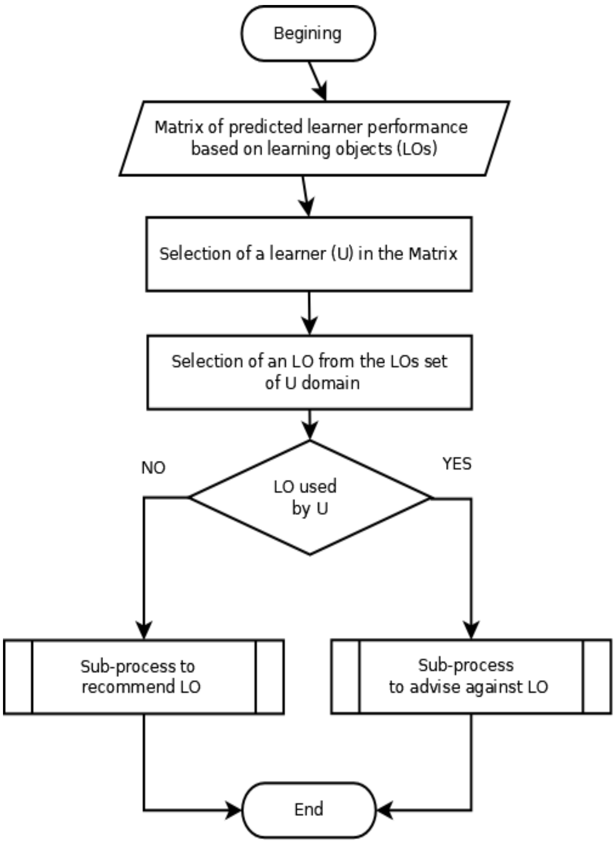


Fig. 1 Learning item recommendation process.

4.1 Add Learning Item Sub-Process

In Fig. 2, the sub-process "recommend learning objects" is presented. It uses as input the learning objects that have not yet been recommended to the selected learner. For a selected learning object, the set of learners similar to the selected learner is determined. Subsequently, the set of learners similar to the selected learner and who have already used the selected learning object, with good predicted performance, is also determined. A recommendation decision-making rate within the interval [0.1] is calculated. This rate is equal to the quotient of the number of learners similar to the selected learner who have already used the selected learning object, with good performance predicted on the number of learners similar to the selected learner. The rate is better and better when it tends towards 1. It is bad when it tends more towards 0. The learning element is recommended if the calculated rate is above a set threshold, closer to 1 than 0.

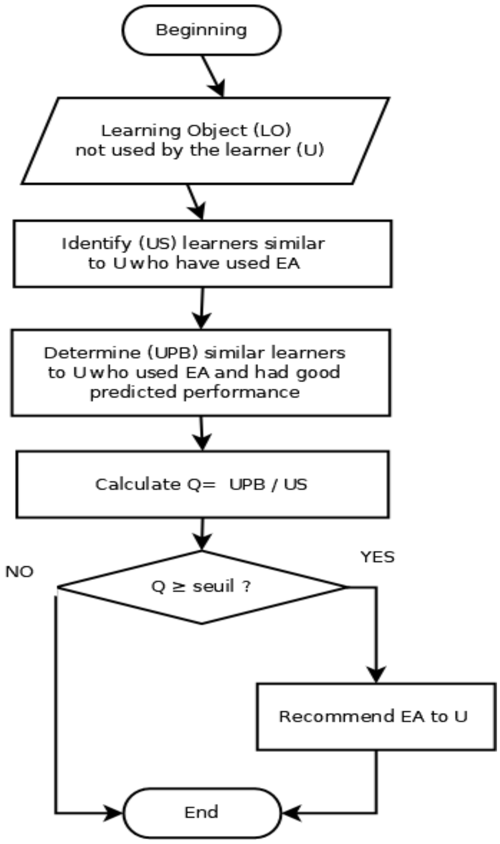


Fig. 2 Sub-process to recommend learning items.

4.2 Learning Item Removal Sub-Process

Fig. 3 shows the "remove learning objects" sub-process. It takes as input learning objects used by the selected learner. An object is selected among these objects. If the learner's predicted performance on this learning object is good, then the object is likely to contribute favorably to his learning. Otherwise, all learners similar to the learner are selected who have already used the selected learning object. Subsequently, it is determined the set of learners similar to the learner, who have already used the selected learning object and having predicted good performance on the learning object. The recommendation decision making rate is calculated. This rate is equal to the quotient of the number of learners similar to the selected learner who have already used the selected learning object, with good performance predicted on the number of learners similar to the selected learner. If the rate is greater than or equal to the set threshold, then the learning object should not be removed from the learner's recommendation list. Otherwise, the learning object is removed. It should be noted that the threshold is set according to the objectives to be achieved. In our work, the threshold has been set at 0.7 because we think it is a good threshold out of 1. Some decision makers may raise the threshold above our choice if they want to make the recommendations even deeper. The threshold will be set below ours if the objective is to make the recommendations lighter.

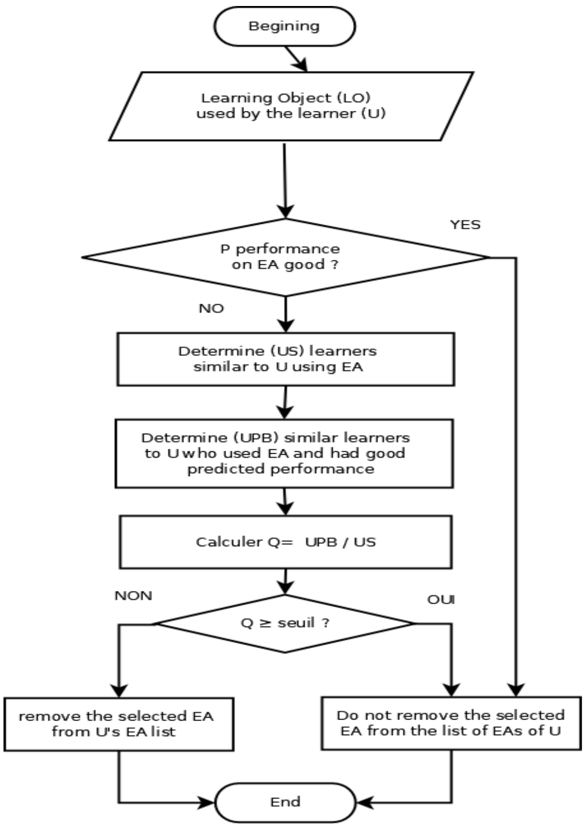


Fig. 3 Sub-process to advise against learning items.

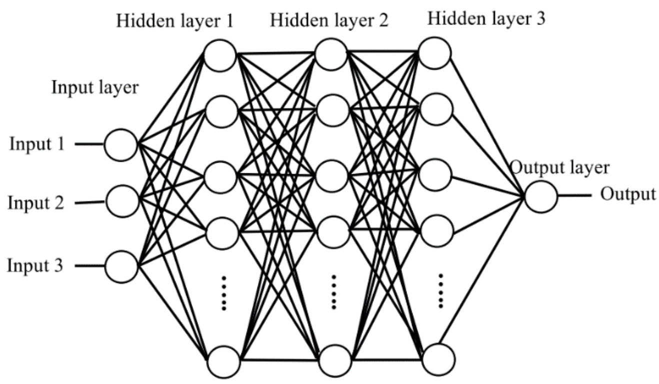
4.3 Dataset

The dataset submitted to our study is that of OULAD (Open University Learning Analytics). This dataset includes students and courses anonymized by numbers and letters respectively. There are seven (7) courses, chosen by each learner. There are 28785 learners. It can be consulted on the following link with its full description: https://analyse.kmi.open.ac.uk/open_dataset. The data set for the experiments is divided into two parts. One part for training data and the other part for test data of 80% and 20% respectively.

4.4 Configuration of our Deep Learning model for performance predictions

The prediction model proposed in this document in Fig. 4 is a Deep Learning with one (01) Input layer, three (03) hidden layers and one (01) Output layer. The Input layer comprises (03) three neurons. The first, second and third hidden layers consist of twenty-eight (28), twelve (12) and six (06) neurons respectively. Input1, Imput2 and Input3 are respectively the number of attempts, the date of submission and the number of consultations of the learning items. Learning is supervised according to Output, which is the learner's score. Deep Learning also takes into account the following configurations: the neuron activation function is 're-read', the solver used is 'adam' and the maximum number of iterations is five (05).

Fig. 4 Deep Learning Model for Performance Predictions.



5. Experimentation, Results And Discussion

5.1 Experimentation

The experiment in this work was carried out with the python programming language, using a four-core I5 computer. This computer operates with a processor speed of 1.70 to 2.40 and a 12 GB RAM memory. Performance predictions were made on the students. As a result of these predictions, the recommend process was performed with the goal of recommending or advising against learning items.

5.2 Evaluation

The evaluation of the experiments undertaken for performance prediction is done with the root mean square error and the confusion matrix. The following formula gives a description of the quadratic error:

$$RMSE = \sqrt{\frac{\sum_{ui}(r_{ui} - \hat{r}_{ui})^2}{n}} \tag{3}$$

The definition of the confusion matrix is made with the help of the following metrics in equations (4), (5), (6) and (7):

Accuracy: refers to the proportion of correct predictions;

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

Precision: refers to the proportion of correct predictions among positive predictions;

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

Recall: the proportion of positives that have been correctly identified;

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

F-Measures: Allows you to evaluate a compromise between recall and precision.

$$F - Measures = \frac{2 * Precision * Rappel}{Precision + Rappel} \tag{7}$$

Table 1: Confusion matrices

		Detected	
		Positive	Negative
Actual	Positive	True positive (TP)	False negative (FN)
	Negative	True positive (FP)	True negative (TN)

TP: positive class predicted correctly;
TN: negative class predicted correctly;
FP: positive class predicted incorrectly;
FN: negative class predicted incorrectly.

5.3 Results of the prediction process

Table 2: Prediction algorithm comparison results

Algorithms	Accuracy	Precision	Recall	F- Mesure
Deep Learning	97.28 %	97.31 %	99.98 %	98.51 %
k-NN	97.03 %	97.20 %	99.57 %	98.37 %

Table 3: Algorithm errors

Algorithmes	RMSE
Deep Learning	0,0251
k-NN	0,0296

5.4 Results of the processes to recommend and advise against

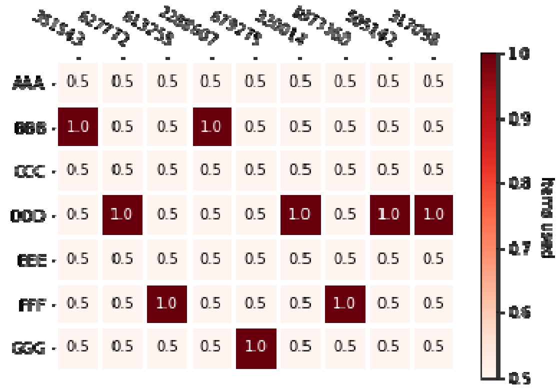


Fig. 5 Sub-process to advise against learning items.

The experience produced results for the learners. Fifty (50) of them were randomly chosen for the discussion phase. They each have items to recommend or advise against. Among these students, the results for nine (09) students are presented in Fig. 6, Fig. 7 and Fig. 8. The choice of presentation of these nine (09) students, as each of them has at least one recommended or advise against items. The remaining forty-one (41) students obtained either recommended or advise against items. Fig. 5 shows the previous states of the nine (09) students according to the learning elements in their profiles.

Fig. 6 Results of learning item recommendations for a threshold greater than 0.9620

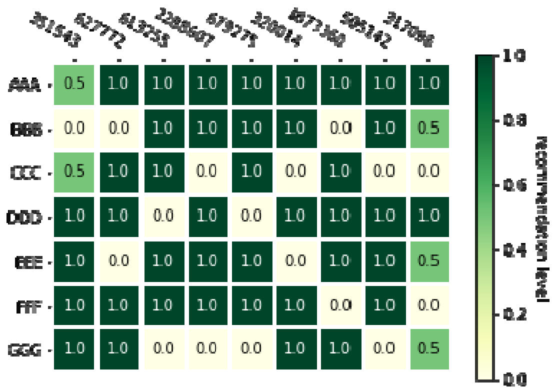


Fig. 7 Results of learning item recommendations for a threshold greater than 0.9720

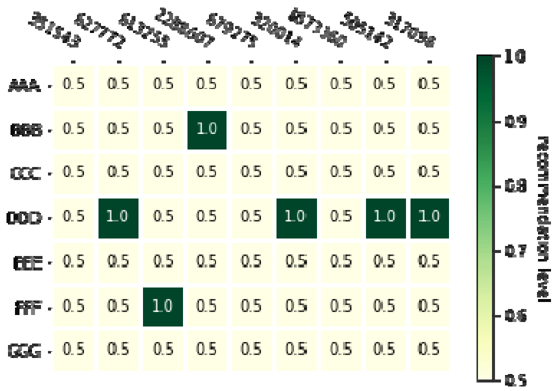


Fig. 8 Results of recommendations for learning items using the Bourkoukou and Achbarou model

Table 4: Prediction algorithm comparison results

	351543	627772	613255	2288607	679275	320014	1873360	505142	317096
AAA	NAN	NAN	NAN	NAN	NAN	NAN	NAN	NAN	NAN
BBB	0.0	NAN	NAN	**1.0**	NAN	NAN	NAN	NAN	NAN
CCC	NAN	NAN	NAN	NAN	NAN	NAN	NAN	NAN	NAN
DDD	NAN	**1.0**	NAN	NAN	NAN	1.0	NAN	1.0	1.0
EEE	NAN	NAN	NAN	NAN	NAN	NAN	NAN	NAN	NAN
FFF	NAN	NAN	1.0	NAN	NAN	NAN	0.0	NAN	NAN
GGG	NAN	NAN	NAN	NAN	0.0	NAN	NAN	NAN	NAN

5.5 Discussions

5.5.1 Discussions On Prediction Results

Table2 presents the results of learner performance predictions for the k-NN and Deep Learning algorithms. The k-NN algorithm is used in the works with which we compare our proposed approach [15]. This work by Bourkoukou and Achbarou is the benchmark in the field of recommendation systems for learning items based on predictions of learner performance. Deep Learning is used for performance predictions in the model proposed in this article. These results show better performance for the Deep Learning algorithm with each of the accuracy, precision, recall and F-Measure metrics of 97.28%, 97.31%, 99.98% and 98.51% respectively. For these metrics, k-NN presents 97.03%, 97.20%, 99.57% and 98.37% respectively.
Table 3 shows the errors made in performance predictions. Deep Learning has less error than k-NN, 0.0251 and 0.0296 respectively.
Thus, Deep learning gives better performance prediction results compared to the k-NN algorithm.

5.5.2 Discussions On The Results Of The Recommendations

In Fig. 5 the learners are anonymized by numbers and the learning items by letters. The cells marked with the color red and the number 1.0 are the learning items used by the learners. For example, learner 613255 uses the learning item 'FFF'. For example, learner 613255 uses the learning item 'FFF'. The cells marked with the number 0.5 with the very light red color represent the learning elements not used by the learners. For example, no learner uses the learning element 'AAA'.

As for Fig. 6 and Fig. 7, the cells marked with the color pure green and the number 1.0 are the learning items recommended for learners. For example, in Fig. 6, the learning items 'DDD', 'EEE', 'FFF' and 'GGG' are recommended for learner 351543. The cells marked with the number 0.0 with the color white represent learning items that are not recommended for learners. For example, learning item 'BBB' is not recommended for learner 2288607. Light green cells marked with the number 0.5 represent neutral learning items, without appreciation for learners.
Fig. 6 shows the results of the recommend and advise against when the prediction values are greater than 0.9620. Fig. 7 shows the results when the prediction values are greater than 0.9720.
Fig. 8 presents the simulation results of the approach with which we compare our results [15]. Cells marked with the color pure green and the number 1.0 are recommended learning items for learners. Cells marked with light yellow and the number 0.5 represent neutral learning items, without appreciation for learners.
Table 4 shows the actual values of the expected predictions. Values 1.0 represent the learning elements that should be recommended to learners. The 0.0 values represent learning items that should be discouraged to learners. The "NAN" indicate that no actual values were expected. Thus,

a recommendation system that aims to be efficient must not only suggest exactly the recommendations of the values 1.0 and 0.0, but also offer other learning items to users, apart from what they are already using. Taking into account the learning items used by learners in Table 4, which presents six (06) items to recommend and three (3) items to discourage, Bourkoukou and Achbarou's approach [15] gives better results than the results of our work in Fig. 6 for recommendations. Indeed, for these results, the prediction threshold is set at performance greater than or equal to 0.9620. Bourkoukou and Achbarou recommend exactly 6 out of 6 expected good recommendations, while our approach for 0.9620 recommends 4 out of 6. However, for what we should advise against, it is only our approach that takes them into account. Thus, he advises against the three items of learning to be discouraged. In view of the above, it can be stated that our approach provides better results than Bourkoukou and Achbarou's approach, if we take into account all the items to be recommended and advise against.

Moreover, by setting the threshold for predictions greater than or equal to 0.9720, our approach provides better results. Indeed, it presents 6 good recommendations out of 6 planned and 3 learning elements to be discouraged out of 3.
Taking the analysis further, on the learning elements not used by the learners, it can be seen that our approach makes appreciative suggestions. In view of the previous performance analysis, the indications of recommendations are more precise for the 0.9720 alone than for the 0.9620 alone and that of Bourkoukou and Achbarou.

It will be necessary to mean the positive impact of the performance prediction when it is more and more precise. This is evident with the difference in the recommendations in Fig. 6 and Fig. 7 for fixed prediction values greater than 0.9620 and 0.9720 respectively. The prediction model provides better recommendations for the set threshold above 0.9720 than for the set threshold above 0.9620. Thus, the more the predicted performance is improved, the better the recommendations presented.

5. Conclusion

This paper focused on the search for a quality recommendation algorithm in the field of e-learning. It was about finding good strategies capable of making the recommendation algorithms perform. In order to give an answer to the stated objective, two axes have been explored. First, the proposal of a recommendation algorithm based on the prediction of learner performance. Then the proposal of an algorithm to advise against learning elements that are not appropriate for the learner. At the end of the work carried out, a dual-function algorithm is proposed. The first function makes it possible to recommend appropriate learning elements for the training of learners. The second makes it possible to advise against learning elements that would be unsuitable for the profile and training of a learner. These two functions perform well with the use of predicted learner performance. These predictions are made using convolutional neural network (CNN) algorithms. They are better than the predictions of the k-nearest neighbor (k-NN) algorithm, which are very good predictors. Enhance the recommendations of e-learning referral systems positively impacting learners' choices. Thus, learners will have the best choices for a fast and quality training. Therefore, these improvements can be of great help to educational administrations and teachers in their decisions and recommendations. In the quest for the continuous improvement of recommendation systems, our future work will aim to advise against learning elements that are not found in a learner's profile.

Acknowledgment

At the location of the Unité de Formation et de Recherche des Mathématiques et Informatique (UFR-MI) of the Université Félix Houphouët-Boigny (UFHB) and the Laboratoire de Recherche en Informatique et Télécommunication (LARIT).

References

[1] J. Delporte, « Factorisation matricielle, application à la recommandation personnalisée de préférences », thesis, Rouen, INSA, 2014.
[2] S. M. Nafea, F. Siewe, et Y. He, « A Novel Algorithm for Course Learning Object Recommendation Based on Student Learning Styles », in 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), févr. 2019, p. 192-201, doi: 10.1109/ITCE.2019.8646355.
[3] J. Lu, D. Wu, M. Mao, W. Wang, et G. Zhang, « Recommender system application developments: A survey », Decision Support Systems, vol. 74, p. 12-32, juin 2015, doi: 10.1016/j.dss.2015.03.008.
[4] J. K. Tarus, Z. Niu, et D. Kalui, « A hybrid recommender system for e-learning based on context awareness and sequential pattern mining », Soft Comput, vol. 22, no 8, p. 2449-2461, avr. 2018, doi: 10.1007/s00500-017-2720-6.
[5] B. Alhijawi, G. Al-Naymat, N. Obeid, et A. Awajan, « Novel predictive model to improve the accuracy of collaborative filtering recommender systems », Information Systems, vol. 96, p. 101670, févr. 2021, doi: 10.1016/j.is.2020.101670.
[6] D. D. A. S. L. Koffi, T. N'takpe, A. Adje, et S. Oumtanaga, « A New Approach to Predicting Learner Performance with Reduced Forgetting », International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, no 5, Art. no 5, 41/30 2020, doi: 10.14569/IJACSA.2020.0110532.
[7] S. B. Aher et L. M. R. J. Lobo, « Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data »,

Knowledge-Based Systems, vol. 51, p. 1-14, oct. 2013, doi: 10.1016/j.knosys.2013.04.015.

[8] S. B. Aher, « EM&AA: An Algorithm for Predicting the Course Selection by Student in e-Learning Using Data Mining Techniques », J. Inst. Eng. India Ser. B, vol. 95, no 1, p. 43-54, janv. 2014, doi: 10.1007/s40031-014-0074-3.

[9] H. N. T. Diem, « Personalized Course Recommendation in Formal Learning Based on Logistic Regression », 2015, doi: 10.17148/IJARCCE.2015.410119.

[10] A. Al-Badarenah et J. Alsakran, « An Automated Recommender System for Course Selection », ijacsa, vol. 7, no 3, 2016, doi: 10.14569/IJACSA.2016.070323.

[11] H. Imran, M. Belghis-Zadeh, T.-W. Chang, Kinshuk, et S. Graf, « PLORS: a personalized learning object recommender system », Vietnam J Comput Sci, vol. 3, no 1, p. 3-13, févr. 2016, doi: 10.1007/s40595-015-0049-6.

[12] J. Xiao, M. Wang, B. Jiang, et J. Li, « A personalized recommendation system with combinational algorithm for online learning », J Ambient Intell Human Comput, vol. 9, no 3, p. 667-677, juin 2018, doi: 10.1007/s12652-017-0466-8.

[13] J. Shu, X. Shen, H. Liu, B. Yi, et Z. Zhang, « A content-based recommendation algorithm for learning resources », Multimedia Systems, vol. 24, no 2, p. 163-173, mars 2018, doi: 10.1007/s00530-017-0539-8.

[14] X. Liu, « A collaborative filtering recommendation algorithm based on the influence sets of e-learning group's behavior », Cluster Comput, vol. 22, no 2, p. 2823-2833, mars 2019, doi: 10.1007/s10586-017-1560-6.

[15] O. Bourkoukou et O. Achbarou, « Weighting based approach for learning resources recommendations », JOIV : International Journal on Informatics Visualization, vol. 2, no 3, Art. no 3, avr. 2018, doi: 10.30630/joiv.2.3.124.

[16] O. Bourkoukou et E. E. Bachari, « Toward a Hybrid Recommender System for E-learning Personnalization Based on Data Mining Techniques », JOIV : International Journal on Informatics Visualization, vol. 2, no 4, Art. no 4, août 2018, doi: 10.30630/joiv.2.4.158.

[17] S. Asadi, S. M. B. Jafari, et Z. Shokrollahi, « Developing a Course Recommender by Combining Clustering and Fuzzy Association Rules », Journal of AI and Data Mining, vol. 7, no 2, p. 249-262, avr. 2019, doi: 10.22044/jadm.2018.6260.1739.

[18] H. Zhang, T. Huang, Z. Lv, S. Liu, et H. Yang, « MOOCRC: A Highly Accurate Resource Recommendation Model for Use in MOOC Environments », Mobile Netw Appl, vol. 24, no 1, p. 34-46, févr. 2019, doi: 10.1007/s11036-018-1131-y.

[19] C. K. Pereira, F. Campos, V. Ströele, J. M. N. David, et R. Braga, « BROAD-RSI – educational recommender system using social networks interactions and linked data », J Internet Serv Appl, vol. 9, no 1, p. 7, mars 2018, doi: 10.1186/s13174-018-0076-5.

[20] Z. Gulzar, A. A. Leema, et G. Deepak, « PCRS: Personalized Course Recommender System Based on Hybrid Approach », Procedia Computer Science, vol. 125, p. 518-524, janv. 2018, doi: 10.1016/j.procs.2017.12.067.

[21] S. Wan et Z. Niu, « A Hybrid E-Learning Recommendation Approach Based on Learners' Influence Propagation », IEEE Transactions on Knowledge and Data Engineering, vol. 32, no 5, p. 827-840, mai 2020, doi: 10.1109/TKDE.2019.2895033.

[22] S. M. Nafea, F. Siewe, et Y. He, « On Recommendation of Learning Objects Using Felder-Silverman Learning Style Model », IEEE Access, vol. 7, p. 163034-163048, 2019, doi: 10.1109/ACCESS.2019.2935417.

[23] Y. LeCun, Y. Bengio, et G. Hinton, « Deep learning », Nature, vol. 521, no 7553, Art. no 7553, mai 2015, doi: 10.1038/nature14539.

[24] N. Rusk, « Deep learning », Nature Methods, vol. 13, no 1, Art. no 1, janv. 2016, doi: 10.1038/nmeth.3707.

[25] L. Deng et D. Yu, « Deep Learning: Methods and Applications », Found. Trends Signal Process., vol. 7, no 3-4, p. 197-387, juin 2014, doi: 10.1561/2000000039.

[26] H. Jiang, « Non-Asymptotic Uniform Rates of Consistency for k-NN Regression », AAAI, vol. 33, no 01, Art. no 01, juill. 2019, doi: 10.1609/aaai.v33i01.33013999.

[27] D. Kurniadi, E. Abdurachman, H. L. H. S. Warnars, et W. Suparta, « The prediction of scholarship recipients in higher education using k-Nearest neighbor algorithm », IOP Conf. Ser.: Mater. Sci. Eng., vol. 434, p. 012039, déc. 2018, doi: 10.1088/1757-899X/434/1/012039.

[28] A. Swetapadma et A. Yadav, « A novel single-ended fault location scheme for parallel transmission lines using k-nearest neighbor algorithm », Computers & Electrical Engineering, vol. 69, p. 41-53, juill. 2018, doi: 10.1016/j.compeleceng.2018.05.024.

[29] N. Maleki, Y. Zeinali, et S. T. A. Niaki, « A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection », Expert Systems with Applications, vol. 164, p. 113981, févr. 2021, doi: 10.1016/j.eswa.2020.113981.

[30] F. Wang, Z. Liu, et C. Wang, « An improved kNN text classification method », International Journal of Computational Science and Engineering, vol. 20, no 3, p. 397-403, janv. 2019, doi: 10.1504/IJCSE.2019.103944.

Dagou Dangui Augustin Sylvain Legrand Koffi received his Master's degree in databases and software engineering at the Unité de Formation et de Recherche en Mathématiques et Informatique (UFRMI) at the Université Félix Houphouët-Boigny (UFHB) in 2016. Currently, he is a PhD student in computer science at (UFHB) in Côte d'Ivoire. His research interests include recommendation systems in e-learning, artificial intelligence and Data Science.

Nouho Ouattara received his M.S. degree in Computer Science in 2014 from the University Nangui Abrogoua in Abidjan, Côte d'Ivoire. He obtained his PhD in Computer Science in 2020 at the Institut national polytechnique Félix Houphouët Boigny, Côte d'Ivoire. He is currently a teacher-researcher at Alassane Ouattara University. He is also a member of LARIT. His current research focuses on artificial intelligence and image processing.

Digrais Moïse Mambe received the PhD degrees from INPHB in 2019. He works as an assistant professor in the Dept. of Informatic, the Nangui Abrogoua University (from 2020). His research interest includes Quality of information and their application to E-learning and Social Web Platforms. He is a member of LARIT Abidjan, Côte d'Ivoire.

Souleymane Oumtanaga received his PhD in Computer Science at Université Paul Sabatier (Toulouse, France) in 1995. He is full professor in computer science at INP-HB since 2007. He was the head of the Network Information Center of Côte d'Ivoire. His research interests include IPv6, wireless network and optical network.

Assohoun Adje received his PhD in Nonlinear analysis at Université catholique de Louvain in 1987. He is full professor in computer Nonlinear analysis at the Unité de Formation et de Recherche en Mathématiques et Informatique (UFRMI) at the Université Félix Houphouët-Boigny (UFHB). He is Director of UFRMI. His research interests include Numerical analysis, EDP, Mathematical Modeling and Computer Science.