

Article

AISAR: Artificial Intelligence-Based Student Assessment and Recommendation System for E-Learning in Big Data

Wala Bagunaid , Naveen Chilamkurti  and Prakash Veeraraghavan 

Computer Science and Information Technology, La Trobe University, Bundoora, VIC 3086, Australia

* Correspondence: 20197186@students.latrobe.edu.au

Abstract: Educational systems have advanced with the use of electronic learning (e-learning), which is a promising solution for long-distance learners. Students who engage in e-learning can access tests and exams online, making education more flexible and accessible. This work reports on the design of an e-learning system that makes recommendations to students to improve their learning. This artificial intelligence-based student assessment and recommendation (AISAR) system consists of score estimation, clustering, performance prediction, and recommendation. In addition, the importance of student authentication is recognised in situations in which students must authenticate themselves prior to using the e-learning system using their identity, password, and personal identification number. Individual scores are determined using a recurrent neural network (RNN) based on student engagement and examination scores. Then, a density-based spatial clustering algorithm (DBSCAN) using Mahalanobis distance clustering is implemented to group students based on their obtained score values. The constructed clusters are validated by estimating purity and entropy. Student performance is predicted using a threshold-based MapReduce (TMR) procedure from the score-based cluster. When predicting student performance, students are classified into two groups: average and poor, with the former being divided into below- and above-average students and the latter into poor and very poor students. This categorisation aims to provide useful recommendations for learning. A recommendation reinforcement learning algorithm, the rule-based state-action-reward-state-action (R-SARSA) algorithm, is incorporated for evaluation. Students were required to work on their subjects according to the provided recommendations. This e-learning recommendation system achieves better performance in terms of true-positives, false-positives, true-negatives, false-negatives, precision, recall, and accuracy.

Keywords: artificial intelligence; clustering; map-reduce; recommendation system; student assessment



Citation: Bagunaid, W.; Chilamkurti, N.; Veeraraghavan, P. AISAR: Artificial Intelligence-Based Student Assessment and Recommendation System for E-Learning in Big Data. *Sustainability* **2022**, *14*, 10551. <https://doi.org/10.3390/su141710551>

Academic Editors: Gwo-Jen Hwang, Morris JONG Siu-yung, Haoran Xie and Gary Cheng

Received: 30 June 2022

Accepted: 19 August 2022

Published: 24 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In traditional learning management systems (LMSs), students and teachers interact with each other in a personal way to gain knowledge. The Internet has transformed many different fields, including education. Most educational institutes have begun to use e-learning due to its flexibility [1–3]. E-learning enables students to access assignments, exercises, and lecture notes, etc., online. It is important to assess student satisfaction with the growth of their education through e-learning. Educational institutions take advantage of the positive aspects of e-learning. The intention to use e-learning is modelled by hypotheses based on self-efficacy, e-learning content, student satisfaction, and perceived usefulness [4,5]. E-learning is an interesting way to improve a student's knowledge since online assistance has recently become the most sophisticated solution.

Currently, e-learning is performing better than traditional education systems [6–9]. Personalised recommendation systems in e-learning have attracted wide attention for improving student learning rates by evaluating their performance. The development of an e-learning system involves certain challenges [10–12]. The major challenges that have been identified in e-learning are as follows:

- The absence of an accurate prediction of the performance of students with certain difficulties leads to a lack of optimal recommendations for them.
- There is a lack of fast-performing algorithms due to the increasing amounts of input data, which affect the exact prediction of student performance [13].

E-learning needs a complete recommendation system to better assist students. To overcome the challenges, artificial intelligence and machine learning methods are used [14–18]. Deep learning algorithms perform faster and attain more accurate results. The advancement of artificial intelligence brings exciting opportunities to education due to its wide range of technologies, features, and functions. AI in education has the potential to provide predictive models, identify high-performing and at-risk students, track academic progress, and design and implement individualised lesson plans, tests, and feedback [19]. By using algorithms, AI can improve learning and make learning more personalised. For example, intelligent tutoring systems (ITS) are a new way of using technology to help people improve their learning. AI can reduce the amount of time teachers/lecturers spend grading and assessing multiple-choice tests and homework, and it can also be used to evaluate essays. There has been increased interest in the use of and potential use of AI in education since the COVID-19 pandemic [20]. The benefit of using artificial intelligence and deep learning algorithms is that they can learn from processing and are also capable of performing parallel processing with massive amounts of data. As a result, the system's performance is robust and scalable, and a learning system driven by big data and supported by AI can make learning more engaging, encourage students to study actively, and enhance academic performance [21].

This paper presents an artificial intelligence-assisted student-collaborative filtering recommendation system based on accurate performance prediction and reinforcement learning algorithms. The scores of individual students are computed using a recurrent neural network (RNN) followed by clustering. Based on the students' performance, they are clustered to make top-performing students stand out from others. The top-ranked students will not receive any suggestions to improve their knowledge since they are already performing well. The performance of the average and poor students is determined using MapReduce. Lastly, reinforcement learning is applied to provide appropriate recommendations for the average and poor students. The performance of this recommendation system is evaluated. The computation of student scores and performance is processed in parallel; hence, the system is efficient in terms of time consumption.

1.1. Motivation

A learning management system (LMS) can track student performance; manage elements of system administration such as user registration, content, calendars, and user access; and provides a way for educators to deliver content over the Internet and across multiple applications, such as through web searching, online tests, online courses, and online e-books, among others. A recent development in e-learning is the inclusion of a recommendation system. In general, a recommendation system in e-learning provides appropriate guidance or suggestions for a particular student to improve their efficiency in a particular field of study. Hybrid recommendation systems such as content-based filtering (CBF), collaborative filtering (CF), and hybrid filtering (HF) were designed in [22–24]. These systems incorporate hybrid methods of filtration and similarity measurements. By analyzing individual academic records and behavioral activity, students are classified as beginner, intermediate, or master. An adaptive feedback system has been incorporated into LMSs, which is essential to measure the varying performance of students. The utilization of artificial intelligence in this system solves this critical issue, as it can make decisions based on student performance [25].

The requirements for a recommendation system are as follows:

- The system should effectively analyse a student's current academic performance and activities and provide an accurate recommendation to improve their performance.

- With an increase in the number of students, the system should make faster computations with zero errors, or else student access will be minimised.
- The system should reduce unauthorised student access, which reduces the number of unnecessary computations that need to be made and connects poor student scores with their better-performing peers.

This research develops an artificial intelligence-assisted e-learning recommendation system. Figure 1 provides an overview of the student recommendation system. The e-learning system is designed with two aims: (i) to predict student performance and (ii) to provide recommendations to average and poor learners.

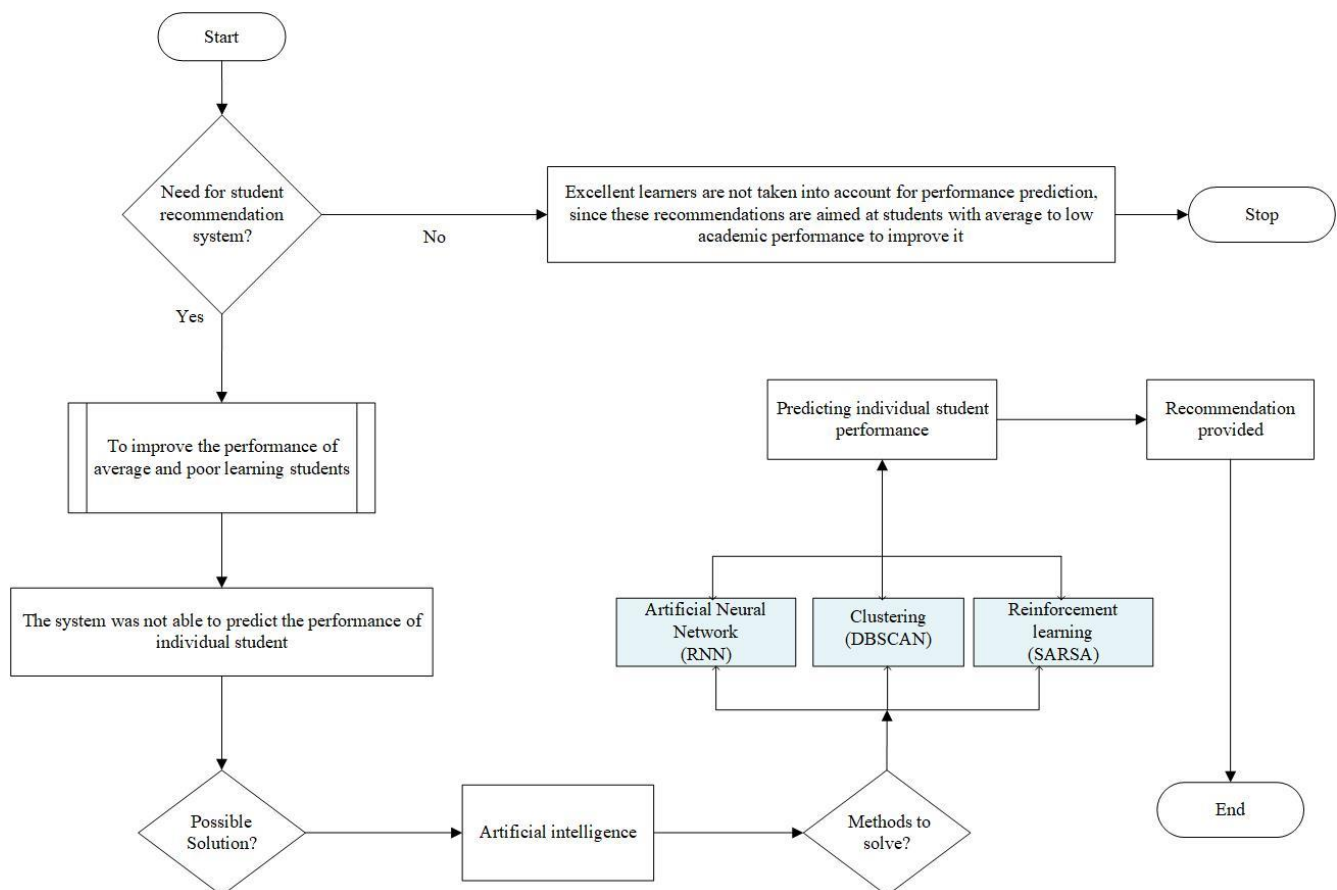


Figure 1. Recommendation system flowchart.

The following questions will be addressed in this study:

1. In e-learning, how can artificial intelligence techniques be combined with reinforcement learning techniques to achieve faster processing and more accurate results?
2. How can we predict the performance of individual students in order to identify students with poor academic performance and enable us to provide appropriate recommendations for them?

To answer these questions, the proposed system is designed to achieve higher accuracy, precision, and recall in predicting a student's performance. The exact prediction of a student's performance enables us to provide appropriate recommendations to individual students.

1.2. Contribution

The significant contributions of this paper are as follows:

- Students' scores are estimated using a recurrent neural network (RNN) that considers both examination results and engagement in the classroom.

- A density-based spatial clustering application with noise (DBSCAN) is applied based on the Mahalanobis distance to extract and classify student performance as excellent, average, or poor.
- To predict the performance of individual students, threshold-based MapReduce (TMR) is applied to average and low-scoring students so that recommendations can be made in a more accurate manner.
- Exact recommendations are presented to students by incorporating reinforcement learning algorithms with artificial intelligence. Rule-based state-action-reward-state-action (R-SARSA) delivers recommendations to students autonomously.

1.3. Organisation

This paper is organised into the following sections: Section 2 describes recent previous works and details their limitations. Section 3 highlights the existence of a major problem, and Section 4 proposes solutions. Section 5 details the implementation setup and compares it to prior work to justify the potential of the proposed result. Finally, Section 6 concludes the research and offers suggestions for future research directions.

2. Related Work

In this section, state-of-the-art works are detailed, including their limitations. This section is divided into two subsections, namely research on student assessment systems and research on e-learning recommendation systems.

2.1. Student Assessment System

In recent years, educational data mining (EDM) has become an effective tool for identifying hidden patterns in educational data and for predicting academic performance. Thus, the academic performance of students has been predicted using different algorithms and methods. The findings support the use of machine learning algorithms to predict student performance [26].

In [27], an LMS stores student activity records, which helps to indicate a student's generic skills. This was incorporated in four subjects. The teachers of these subjects were surveyed using the activity records, and the students' generic skills were evaluated. However, the teacher surveys had to be updated frequently for evaluation purposes, and student records were collected to assess their performance. In [28], student assessment was addressed based on the collected profiles and previous experience. In this work, the Levenshtein distance measure was incorporated to estimate the similarity between students. The activity layout of each student was maintained with their previous score values using the reports of the first six students, which were verified by distance estimation. However, distance similarity was time consuming because of the large number of students.

In [29], student performance was predicted using homework submissions, and students were labelled as either procrastinators or non-procrastinators using a k-means clustering algorithm. Furthermore, the students were classified using the following methods: ZeroR, OneR, ID3, J48, random forest, decision stump, JRip, PART, NBTree, and Prism. As a result, the classifier's output was analysed under different classes. However, the consideration of a single constraint of homework to evaluate a student's assessment was not effective. In [30], an online assessment system for learning and study strategies inventories (LASSI) was developed to assess student skills in a particular course. In-class activities and assignments were observed, and the score values were then determined based on the subject area that corresponded to the students. From these scores, the learning assessment of the student was measured.

In [31], an inductive miner algorithm was modelled to determine pass and fail students according to the estimated fitness values. The four key attributes that were taken into account were time, identity, action, and information. During preprocessing, duplicate records of students and instructors were deleted. For every unit (pass students, fail students, all students), three inductive miners were used for the purposes of determining

the corresponding fitness values. As per the fitness value, the students were categorised as either pass or fail for each unit. This was to identify the performance of all students as opposed to individual performance.

The study in [32] proposed a hybrid multi-attribute decision-making method with fuzzy complex proportional assessment (COPRAS) that was used to rank e-learning websites, which was helpful for students in terms of making a selection. However, the websites needed to be relevant to the students' interests. Hence, selecting a website from a questionnaire or from a set of ratings was not a major part of e-learning. Additionally, these constraints differed based on the courses and the students' study plans.

In [33], clustering was an effective solution, and four agents were designed: a project clustering agent, a student clustering agent, a student–project matching agent, and a student–student matching agent. A dynamic student clustering agent predicts student behaviour. A particle swarm optimisation (PSO) algorithm was used to estimate the fitness value. Based on the computed centroid, the clusters were constructed. Additionally, clustering was performed with k-means, k-medoids, fuzzy C-means, expectation–maximisation, and clustering by fast search and finding of density peaks via heat diffusion (CFSFDP-HD) [34,35]. However, a common issue with clustering is that it fails to predict student performance individually, and it is only capable of determining good and bad performance.

2.2. E-Learning Recommendation System

A key part of artificial intelligence systems in education is the ability to make recommendations. A recommendation system's capabilities are a critical component of AI systems in e-learning. Thus, ensuring student equality in recommended educational opportunities is critical, as the recommended courses may result in educational gains or losses for them. In [36], the authors explored ten of the most advanced recommendation systems in a massive open online course platform and revealed that they provide systematically unequal learning opportunities for students. In [37], an agile-based approach was constructed for collaborative communication between students and professors. The professors were able to share study materials, syllabi, and exercises that would enhance student performance. This type of communication was introduced to provide appropriate suggestions for students. In [38], a recommendation generation system was presented based on self-organisation behaviors. The adaptability and diversity of recommendations were improved by proposing learning object analysis. The three main constraints that were taken into account were learning goals, learning styles, and the learner's behavior. This was not sufficient; hence, a student's current ability is also required.

In [39], a hybrid approach was proposed in which ontology was applied to retrieve the significant information required for recommendations. In this paper, the N-Gram and the query expansion approaches were combined and used to retrieve recommendations based on the learner's query and course database. The recommendations for courses from similar learners were provided to active learners. Recommendation systems require an accurate assessment of student performance. In [40], a parallel frequent pattern (FP)-growth algorithm was developed by evaluating learner behaviour and preferences. The core idea was based on mining frequent item sets using an FP-growth algorithm. The recommendation engine analyses the learner's behaviour and activities to provide appropriate recommendations. However, the frequent extraction from the item sets was based on the frequency of a certain item in the dataset, which was not appropriate for student recommendations.

In [41], a similarity matrix was estimated to evaluate similarity based on the user's information and behaviour. Then, the ranking strategy was applied to the generated recommendation list, which was viewed by all of the other students. However, a recommendation for a particular student should only be able to be viewed by that student, for privacy reasons. In [42], recommendations were made to students based on their knowledge using a machine learning method. However, this work was completed based on latent Dirichlet allocation (LDA), which processes presently used words. In [43], recommendations were made to students after they had undergone the authentication process. An adaptive trust-based

e-assessment system for learning enables authentication for online education students. In this work, authentication in e-learning is based on knowledge, biometrics, possession, and mechanisms. However, despite its importance, authentication in recommendation systems has not received wide attention.

The significant findings that identified from the existing literature are as follows:

- Student performance was evaluated without the provision of recommendations, which are helpful for the student to improve their academic scores in future examinations.
- The evaluations of student performance could not accurately identify the score values of individual students.
- Student authentication was not a focus, which allowed unauthorised students to engage in academic malpractices, which affected performance evaluation.
- The increase in the number of students who engaged with the system required results to be obtained faster and for appropriate recommendations to be provided to improve student performance.

3. Problem Statement

This section presents the research on student assessment. In [44,45], machine learning algorithms and deep learning methods were used to estimate student scores. A multimodal assessment framework was designed with CNN and was based on latent semantic analysis (LSA), and machine learning algorithms such as decision tree, J48, gradient-boosted tree, CART, and the Naïve Bayes algorithm were used. However, these methods only achieved accuracy rates of 88% or less. Additionally, writing was taken into account as a risk factor when predicting student performance.

In [46], decision tree-based classification was utilised for the estimation of a belong matrix based on a mass vector. The construction of tree structures for classification cause major changes in the tree when new data are included. Therefore, there will be a reduction in the system's accuracy during classification. Clustering, machine learning, and deep learning were used to handle this problem in a priori works, resulting in a set of predication problems. The problems were as follows:

- There was an absence of recommendations for the students, which are essential for them to improve their knowledge.
- Grouping students based on their performance (average, good, poor) is not effective in predicting the knowledge level of students individually since the suggestions for instructors will change based on student performance, i.e., not all of the average students require similar recommendations for additional exercises or tests.
- There was the presence of unauthorised students because of a lack of security. This increased the system processing time and led to poor performance predictions.

The problematic issues discussed in this section are overwhelming and can be addressed by the proposed solutions that predict accurate student performance and provide appropriate recommendations to improve their knowledge. The next section describes the procedure of the proposed artificial intelligence-based recommendation system.

4. Proposed E-Learning Recommendation System

The proposed system is designed to provide an appropriate recommendation system for students. Student performance is analysed based on the submitted student details, and appropriate recommendations are sent. This system combines student engagement and student scoring features to attain optimal results. Figure 2 illustrates the proposed e-learning recommendation system. As shown, the proposed system involves the computation of scores, clustering, and performance prediction and provides recommendations.

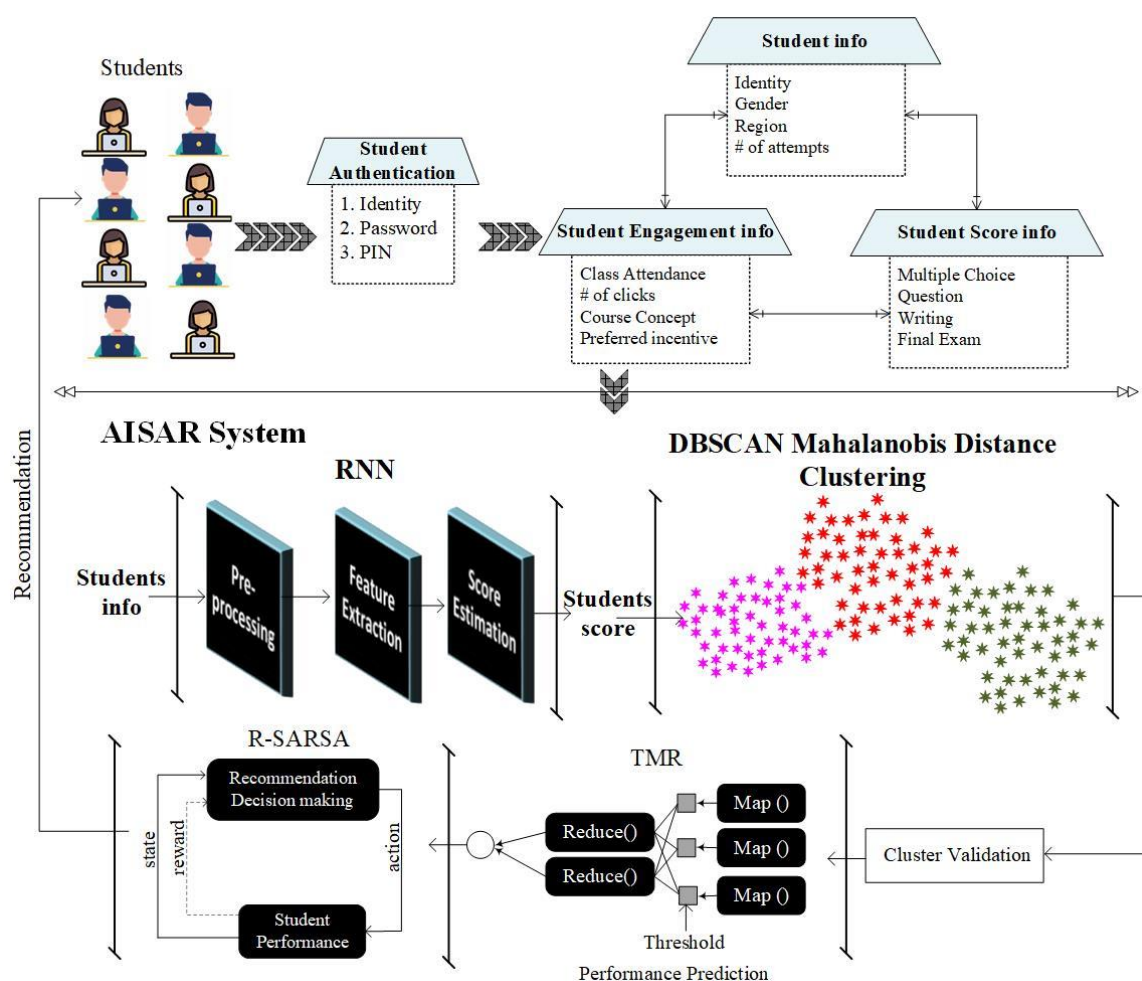


Figure 2. Artificial intelligence-based student assessment and recommendation system (AISAR).

The overall scores of individual students are estimated using RNN, which is composed of preprocessing, feature extraction, and score estimation. Then, using Mahalanobis distance clustering, DBSCAN is employed to cluster students based on their performance in terms of scores. To identify the actual performance of each student, a threshold-based MapReduce process that categorises performance individually is applied. Lastly, recommendations for students are provided by the reinforcement learning algorithm known as SARSA. Each process involved in this system is explained in detail in the following sub-sections.

4.1. Authentication

Some conventional e-learning platforms employ an authentication method that either provides low security at the expense of guaranteed usability or low convenience at the expense of higher security [47].

Unfortunately, with the growth in the use of e-learning, some students engage in academic malpractices to avoid the learning process. Some students share their login information with others so that someone else can complete a test on their behalf to receive higher grades. Student authentication is essential in the e-learning environment to prevent test results that have been obtained by cheating being stored in databases. The risk of online cheating can be reduced by authentication throughout the learning session.

Furthermore, some students who are not enrolled in a course may attempt to access e-learning platforms using their previous log-in credentials.

In the proposed system, the student must authenticate themselves by providing their ID, password, and personal identification number (PIN). The PIN is generated and sent by email after the ID and password have been entered.

Each student is required to submit all of the security credentials correctly. The correctness of each credential is weighted as 1, and their total authentication weight is given as

$$AW = w_1id + w_2pw + w_3pn \quad (1)$$

The terms w_1 , w_2 , and w_3 are the weight values, where $w_1 + w_2 + w_3 = 1$, and AW is the authentication weight that is estimated from identity id , password pw , and PIN pn . If any authentication credentials are incorrect, the corresponding weight value will be 0, and an “Unauthorized Student” message shown in Figure 3 will appear to the user. Hence, students with the highest level of authentication will be allowed to take their tests. The student is authenticated each time they access the system while taking their online tests. A weight value is estimated using the student’s authentication information obtained from these security credentials. Once they complete their tests online, their score values are computed to determine their performance. In light of the student’s performance, they will be sent an appropriate recommendation to improve their academic score on future tests.



Figure 3. Authentication phase.

4.2. Student Score Estimation

One of the significant contributions of machine learning and AI is their capability to predict students’ academic performance using artificial neural networks [48].

In this work, the scores for students are predicted using an RNN that performs preprocessing, feature extraction, and score estimation. The RNN is designed with an input layer, a hidden layer, and an output layer. Information on the student’s engagement and academic score are fed into the input layer. Then, in the hidden layer, preprocessing and feature extraction are performed. Finally, in the output layer, the score value is estimated.

Firstly, preprocessing is performed using data cleaning, and duplicated data are eliminated. The data cleaning process is performed to detect and eliminate error attribute entries present in the dataset. Processing of algorithms without data cleaning leads to unreliable and irrelevant results. Duplicate data are identified and eliminated by computing the Manhattan distance between records.

The Manhattan distance M_D is calculated as follows:

$$M_D = \sum_{i=1}^N |R1_i - R2_i| \quad (2)$$

Let N be the total number of attributes on individual student records $R1, R2, \dots$. If the distance measure $M_D = 0$, then the records are duplicates; otherwise, if $M_D > 0$, the records are not duplicated. After eliminating the duplicate records from the dataset, the

attributes are extracted for score estimation. The features that are extracted from student engagement are class attendance, number of clicks, course concept, and the preferred student incentive, which is awarded to the students upon completion of their activity. The incentive could be points, badges, or other rewards that motivate the student to improve their classroom activity. As per this incentive, the student's activity can be predicted. Then, the score features that are extracted are multiple-choice questions, writing samples, and final examination results. Students with better engagement and who have higher scores will have a higher performance. High-scoring students will have better engagement in the class; if not, their incentives will be very low.

RNN is different from traditional NN since NN is enabled with a memory property. In RNN, the current state is mathematically formulated as

$$C_s = f(C_{t-1}, I_s) \quad (3)$$

The current state C_s is determined based on the function of C_{t-1} and I_s as the previous state and input state, respectively.

In RNN, loops are constructed from the output to the hidden layer in order for the estimated information to persist. This persistence of information is handled in the hidden layer, which memorises the computed information. The RNN for score estimation is depicted in Figure 4, where the processing of each layer is shown.

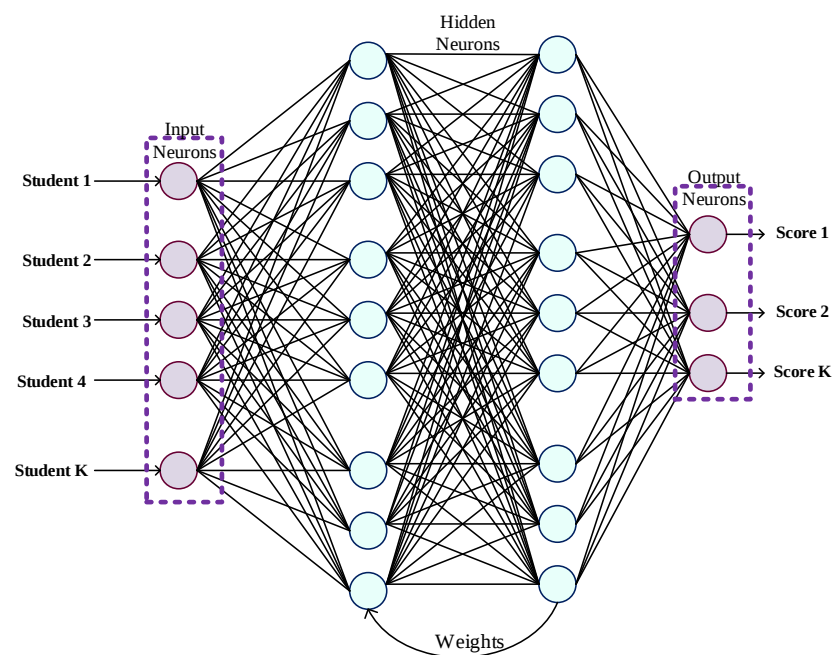


Figure 4. Score estimation in RNN.

The RNN is aware of previous output, and the score values are determined appropriately based on key attributes. In the RNN, the property of memory enables previously calculated information to be stored. The memory of the neurons in the hidden layers effectively reduces the complexity due to the maintenance of the information computed from the input. The property of learning in the RNN is also supported, even with larger sequences. Hence, RNNs are preferred for estimating the score values for the students. The RNN can estimate the score value for the number of students participating in the e-learning system.

4.3. Score-Based Clustering

After determining the score values of the students, they are clustered into three groups: poor, average, and excellent (see Table 1 below). Top-scoring students are clustered as “excellent”. Similarly, based on their score values, they are clustered using the Mahalanobis distance in DBSCAN. The distance measurement reflects the clustering; thus, the Mahalanobis distance is used, which is suitable for similarity measurement. This clustering algorithm is preferred since it can deal with large-sized data. After the clusters are constructed, they are validated by measuring their purity and entropy. If the purity level is lower than the pre-defined threshold value, the cluster is constructed again. During clustering, students will be grouped within certain ranges, within which one’s performance cannot be identified.

Table 1. Score-based DBSCAN clustering.

Score Values from RNN	Score-Based DBSCAN Clustering
Above 80	Excellent
50–80	Average
Below 50	Poor

These clusters are constructed by taking the values of epsilon ϵ and minimum neighbouring points $minPts$ into account. ϵ is the distance that is determined for the data points and is the key to clustering them. The similarity between two data points, i.e., students, is computed using the Mahalanobis distance.

Using this distance measure, all of the neighbouring points existing within epsilon are predicted. DBSCAN clustering is efficient in clustering and is suitable for increasing density, i.e., it can support clusters when e-learning students are included. This DBSCAN is associated with two properties:

ϵ is the distance that is determined for the data points and is the key to clustering them. The similarity between two data points, i.e., students, is computed using the Mahalanobis distance.

Density reachability: Let x_1 be a point that is defined to reach another point, x_2 , that exists in the density within the distance and ϵ . Put simply, the points x_1 and x_2 are supposed to have the number of neighbouring points within ϵ .

Density connectivity: Let x_1, x_2, x_3, x_4 , and x_5 be the data points that are to be clustered. Assume two data points, x_1 and x_2 , are linked by the density and that x_3 is linked with the number of neighbouring points, in which x_1 and x_2 are present within ϵ . Density connectivity is a chaining process, i.e., $x_2 \rightarrow x_3 \rightarrow x_4 \rightarrow x_5 \rightarrow x_1$, that defines that x_2 is a neighbour of x_3 , that x_3 is a neighbour of S , and so forth. This implies that x_2 is a neighbour of x_1 .

The algorithmic steps followed for the construction of DBSCAN clustering are presented in the following pseudo code. Let the data points be represented as $\{x_1, x_2, x_3, x_4, x_5, \dots, x_n\}$, where these points are the score values of individual students. The distance measure plays a vital role in DBSCAN clustering; thus, the Mahalanobis distance was used in this work. Since the clusters are constructed based on the initial prediction of the distance between the points, the cluster members are also added to the distance being evaluated. As a result, DBSCAN requires an appropriate distance measurement, which is enabled by the Mahalanobis distance. This distance formula is relatively better than the conventional Euclidean distance in terms of scalability and correlation. This reflects the improvement in the accuracy of the system.

In DBSCAN (Algorithm 1), the value of ϵ is selected based on the applied Mahalanobis distance measure, which is accurate in cluster formation. Then, the $minPts$ depends on the number of attributes A that are present in the dataset and is given as $minPts \geq A + 1$. The clustered data points are then validated using cluster purity and entropy.

The significant purposes of cluster validation are:

- The effectiveness of the constructed clusters is detected, which enables the system's accuracy to be improved.
- The risk factors are reduced after completing the entire classification process.
- The goodness, i.e., quality, of the constructed cluster can be measured without merging dissimilar data points into a cluster.

Algorithm 1: DBSCAN Clustering Based on Mahalanobis Distance

Input—Data points

Output—Clusters

1. Begin.
 2. Select a i th point as x_i from the x_n data points.
 3. Assign that the data point x_i be visited.
 4. Identify all the neighboring points that are present until the distance ϵ . Let it be denoted as NB_p .
 5. If ($NB_p \leq minPts$)
 - {
 - take x_i as the initial point for creating a new cluster
 - add cluster members as the data points present within distance ϵ into the cluster
 - similarly add members based on NB_p
 - else
 - set the data point x_i as noise
 - }
 6. Repeat steps 1–4 until all the data points are clustered.
 7. End.
-

This work validates the clusters by computing entropy and purity from the clusters created by DBSCAN. The entropy measure is computed for each cluster and is based on the determined class distribution. Furthermore, entropy is estimated for the total number of clusters by applying the addition operator that sums up the clusters. Once the cluster is constructed, its entropy and purity are evaluated to validate the cluster's strength. If the entropy and purity are high, then student performance is predicted from the constructed cluster. If not, the clusters are re-created by estimating the distance between the data points.

4.4. Student Performance Prediction

The actual performance of each student is predicted from the clusters, the students are grouped based on the score values. However, even though the students were grouped, predicting the performance of individual students was not possible. Thus, TMR was incorporated to complete the task. Excellent learners were not considered for performance prediction since the recommendations were only for average and poor learners. The group performance of the average and poor learners was predicted.

MapReduce involves two key steps: mapping and reducing. In this TMR, the threshold value was determined to predict student performance based on student scores, i.e., determined by Shannon entropy.

According to the students' performance, the threshold is predicted and then reduced. This process ensures the accurate prediction of individual student performance. The mapping operation includes splitting and mapping, and the reducing operations include shuffling and reducing. The cluster values are taken as input, and the student's results are estimated. The processes are as follows:

Map: Initially, the input cluster is split into the individual score values of individual students. Then, the map phase is executed as the first phase. Each split value from the cluster is operated on based on the mapping function in this phase. This mapper function is presented by processing the key value pairs that are represented as (k, v) . Let the score values of the student be assigned as key value $k1$ and the student attendance

key value k_2 . As per the k -value, mapping is performed. The output from the mapper function is (k', v') .

Reduce: During shuffling, the mapped output (k', v') values are processed by consolidating the matched records in the mapping phase. This shuffling enables duplicate values to be eliminated, and then it groups similar k -values, which results in $(k, v[])$. The $v[]$ denotes an array of values that is determined from the shuffle operation. During shuffling, the threshold values are fed based on the scoring values. Then, in the reducing phase, the shuffled output is reduced into an output with an exact student performance prediction.

Average students are divided into below average and above average. Similarly, poor students are divided into poor and very poor. MapReduce uses parallel processing, and thus, the results are obtained faster. MapReduce also employs a specific threshold value, which reduces student performance by the exact amount. The pseudocode for TMR highlights the processing of the mapping and reducing functions that map to a student's score value and attendance. In this way, student performance can be predicted (Algorithm 2).

Algorithm 2: TMR Procedure

Let A be the attendance of the student and S be the score value of the student.

Input—Clusters

Output—Student performance prediction

1. Begin.
 2. Initialise cluster 1, cluster 2.
 3. For each cluster, complete the steps below.
 4. Split cluster values.
 5. Function (Map) // start mapping function.
 6. For each k -value.
 7. Extract $(A, (S, 1))$.
 8. Returns (k', v') .
 9. Repeat for each k -value // end mapping function.
 10. Function (Reduce) // start reduce function.
 11. For each $(A, (S, v))$ do

{

compute sum U of $S * v$

find v_{new}

 }
 12. Repeat for each k -value.
 13. Determine $R = U / v_{new}$.
 14. Store $(A, (R, v_{new}))$ // end reducing function.
 15. End.
-

The proposed TMR for predicting student performance based on the clusters is demonstrated in Figure 5. Since this is a parallel processing model, the result of the performance prediction of all the average students is performed one at a time. Similarly, the performance of poor learners is also predicted to be either very poor or poor. This prediction impacts the efficiency of the recommendations for students to improve their academic scores. The student's current performance must be considered when adapting the recommendations, which means that the student's performance has to be predicted accurately.

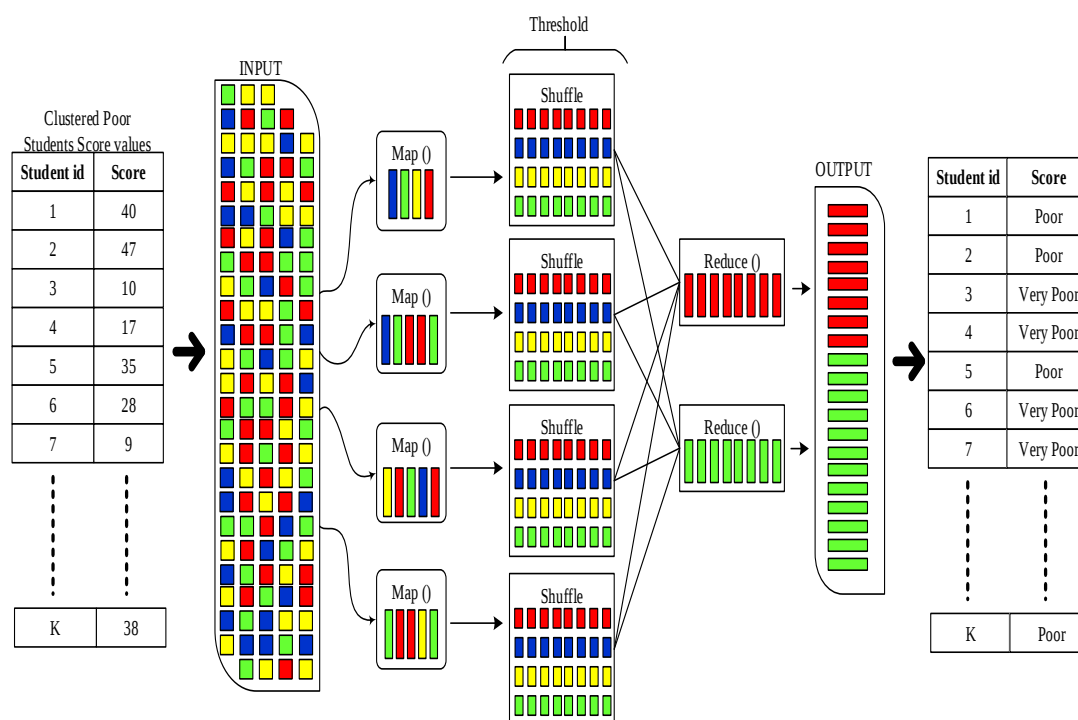


Figure 5. TMR model-based student performance prediction.

4.5. Student Recommendation

The integration of deep neural networks and reinforcement learning has recently gained popularity. When it comes to dealing with high-dimensional states or/and actions, reinforcement learning has become an essential component of future AI systems due to deep neural networks [49]. With the help of an artificial neural network, the system gives students immediate feedback based on what they say. This helps them to gradually understand abstract ideas and complete practical exercises [50]. The recommendations for students were provided using R-SARSA, which is a reinforcement learning algorithm. This reinforcement model can solve difficult problems by learning the environmental conditions. The recommendations for students were presented to improve their learning efficiency. A set of rules was determined from the mean value of student score and engagement. Along with the probability value of the rule, the current performance was also taken into account for recommendations. The states in R-SARSA are defined based on the probability value and current performance.

The rule is defined as shown in Table 2, where the mean values for individual students are computed.

Table 2. Rules defined in R-SARSA.

Rule Number	Input		Assessment Value $p(m)$
	Mean Score	Mean Engagement	
1	x	x	0
2	x	y	0.5
3	y	x	0.5
4	y	y	1

The probability value ranges between 0 and 1. If the mean score is x and is below 0.5, then the probability will be 0. Similarly, y denotes that the obtained mean value is above 0.5. Based on the values of x and y , the probability is determined. The states and actions are defined in the proposed R-SARSA, as shown in Table 3.

Table 3. State–action pairs in R-SARSA.

State	Action	Reward
$s_1 \rightarrow (P(m), S)$	$a_1 \rightarrow$ Practice exercises	r_1
$s_2 \rightarrow (P(m), S)$	$a_2 \rightarrow$ Simple study materials	r_2
$s_3 \rightarrow (P(m), S)$	$a_3 \rightarrow$ Understandable presentations	r_3
$\vdots$	$\vdots$	$\vdots$
$s_n \rightarrow (P(m), S)$	$a_n \rightarrow$ Sample questions	r_4

According to the state estimates for each student, actions are worked out. The actions $\{a_1, a_2, a_3, \dots, a_n\}$ are the recommendations given to students: practice exercises, additional learning courses, and so on. As a result, based on the predicted performance of the student, recommendations are provided.

5. Experiment Result Analysis

The AISAR system and its performance is evaluated in this section. This section consists of three sub-sections: the implementation environment, comparative analysis, and results discussion.

5.1. Implementation Environment

This section details the implementation design of this proposed AISAR system. This system comprises artificial intelligence algorithms that ensure efficient decision making in complex and tedious situations. In other words, the artificial algorithms take a step to learn about the input data to produce an output. The AISAR system was then incorporated to predict student performance and to provide recommendations.

Initially, RNN was used to predict scores; DBSCAN clustering was used to group students based on their score values; TMR was used to predict the performance of individual students; and R-SARSA was used to make recommendations.

This AISAR system was tested in a virtual learning environment (VLE). A VLE is defined as an educational technology that depends on web access, and it opens up the digital aspects of courses for students to complete their academic studies. Here, a VLE-based dataset was used to evaluate this AISAR system. The Open University Learning Analytics dataset (OULAD) [51] was developed for e-learning testing environments. It includes assessment results that were submitted by students as well as detailed VLE data that academics and researchers can use to design different features and create different models for predicting student performance in a course.

The dataset can be used in a variety of scenarios. It allows for the evaluation and comparison of predictive models to predict student assessment results and final course results developed by other researchers. The VLE data allow for the examination of course structure from the standpoint of learning, and the data can be used to assess the impact of VLE on learning outcomes [52].

The OULAD-VLE dataset consists of data from 32,593 students and 22 courses. It also included the daily click activity based on 10,655,280 entries. The dataset was composed of three different data types: demographics, performance, and learning behavior (see Table 4 for dataset details). To ensure privacy, an anonymisation process was performed, and certain attributes were randomised.

Table 4. Dataset details.

Data File	Description	Attributes
Courses	Contains the list of all available modules and their presentations	- code_module - code_presentation-length
Assessments	Contains information about assessments in module presentations. Usually, every presentation has a number of assessments followed by the final exam	- code_module - code_presentation - id_assessment - assessment_type - date-weight
VLE	Contains information about the available materials in the VLE. Students have access to these materials online and their interactions with the materials are recorded.	- id_site - code_module - code_presentation - activity_type - week_from - week_to
Studentinfo	Contains demographic information about the students together with their results.	- code_module - code_presentation - id_student-region - highest_education - imd_band-age_band - studied_credits - disability-final_result
StudentRegistration	Contains information about the time when the student registered for the module presentation. For students who unregistered, the date of unregistration is also recorded.	- code_module - code_presentation - id_student - date_registration - date_unregistration
StudentAssessment	Contains the results of student assessments. If a student does not submit the assessment, no result is recorded. The final exam submissions are missing if the result of the assessments is not stored in the system.	- id_assessment - id_student - date_submitted - is_banked - score
StudentVLE	Contains information about each student's interactions with the materials in the VLE.	- code_module - code_presentation - id_student-id_site - date-sum_click

On the other hand, student-identifying attributes such as gender, disability, age, region, and education were preserved. The dataset used for processing this artificial intelligence-based recommendation system is depicted with attributes that exist in four categories. We extracted three types of data: demographic (code_module, code_presentation, id_student, disability, and studied_credits), performance (final results and score on the assessment), and learning behavior (sum-clicks on VLE activities, num_of_prev_attempts). The attributes were extracted and executed for testing the designed AISAR system, and the dataset attributes are presented in different tables.

Firstly, the scores of the students were computed in the RNN, which was followed by the process of grouping the students. The students were put into groups based on their score values, and the MapReduce programming model used this information to predict their performance.

The key goal of this work was to provide student recommendations, which were attained by a reinforcement learning algorithm that is enabled with the property to learn from the environment.

5.2. Comparative Analysis

This section evaluates the proposed AISAR system results by comparing them with previous research work. This system is compared to previous student engagement predictions in e-learning [45]. This work was also tested using the OULAD dataset by incorporating a machine learning algorithm. Machine learning is popular in different areas; hence, it was applied in e-learning. The algorithms in machine learning are a sub-field of the artificial intelligence field. The AISAR system is compared to the previously existing best-performing machine learning algorithm. The common issues that were identified in the student assessment system are as follows:

- Traditional machine learning algorithms are subjected to critical and problematic limitations such as computations, time consumption, and poor performance prediction.
- Clustering is presented as a solution for predicting student performance. However, this was an effective solution, but it was only able to identify the group performance of the students, i.e., it could not determine the individual performance of a student.
- Student recommendations were not optimal for each student participating in the e-learning system.

Table 5 details the machine learning algorithm used for student score assessment. These algorithms also have significant disadvantages that are highlighted. J48 decision tree algorithms achieve machine learning algorithms with higher accuracy. Machine learning algorithms are extensively used in multiple fields for processing data and retrieving output as per the designed system. Machine learning algorithms are also subjected to certain limitations.

Table 5. Previous work and its demerits.

Work	Methods Used	Disadvantages
Student engagement prediction by machine learning algorithms [45]	Decision tree	• Lower accuracy. • Complex computations. • Instability.
	J48	• Depth of the tree increases execution complexity. • Storage problem due to the growth in depth of the tree.
	CART	• Unstable due to a massive change in the structure for a smaller change in the dataset. • Unable to yield optimal results.
	Gradient boosting tree	• Essential to perform separate tuning process for higher accuracy. • The process for training is slower.
	Naïve Bayes	• Lower accuracy due to independence of classes. • Decrease in precision value when processing smaller-sized datasets.

Existing work [45] was compared to the proposed AISAR system in terms of true-positive rate, false-positive rate, precision, and accuracy. The proposed work incorporates RNN, DBSCAN, TMR, and R-SARSA for score prediction, clustering, performance prediction, and recommendation provisioning. The obtained results are compared in graphical plots that justify the better efficiency of the proposed system.

5.2.1. True-Positive and False-Positive Rate

The true-positive rate and false-positive rate are the key metrics for evaluating the system's performance. The true-positive rate is defined as the correct prediction of student performance. The false-positive rate is defined as the incorrect prediction of student

performance. The mathematical expression for the true-positive rate and false-positive rate T_{PR} and F_{PR} is given as

$$T_{PR} = \frac{T_P}{(T_P + F_N)} \quad (4)$$

$$F_{PR} = \frac{F_P}{(F_P + T_N)} \quad (5)$$

The terms T_P , F_P , T_N , and F_N denote true-positive, false-positive, true-negative, and false-negative, respectively. From these measurements, the true-positive rate and false-positive rate are determined. The positive rate refers to the accurate identification of the correct or the incorrect performance of the student.

The true-positive rate, also known as sensitivity, is measured based on the actual results that are predicted from the dataset. Figure 6 shows an increase in the true-positive rate compared to previous research work using machine learning algorithms. The machine learning algorithms were involved in predicting student performance, which is subjected to certain limitations. In the proposed AISAR system, a student's score is computed based on their engagement and the examinations, according to which they are clustered. Then, the students in each cluster are individually identified using TMR, which accurately maps the score values and then reduces them. The threshold value is provided based on the clustered student performance, so the true-positive rate is higher than that of conventional machine learning algorithms.

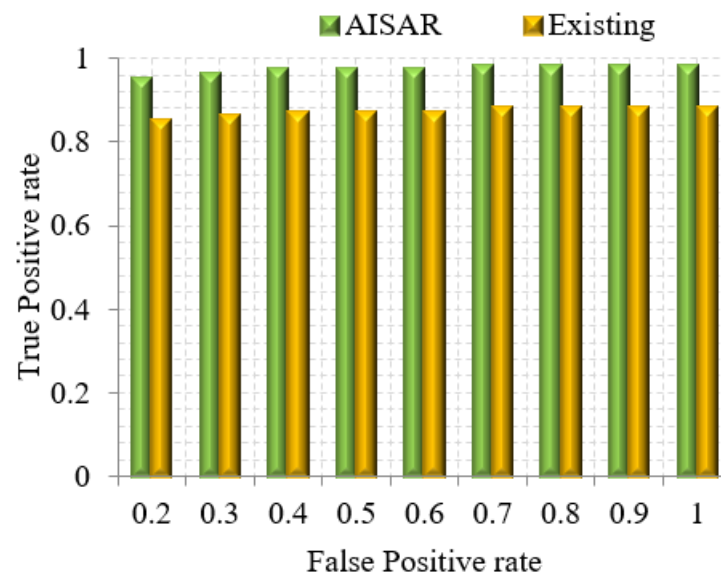


Figure 6. Comparison of false-positive rate versus true-positive rate.

The true-positive rate is 0.97 in the AISAR system and 0.87 in the existing J48 machine learning algorithm. This increase is due to the correctness of the prediction of the student performance, which is based on the exact prediction made using TMR.

5.2.2. True-Negative and False-Negative Rate

The true-negative rate is the correctly predicted student performance based on negative results. The false-negative rate is the incorrectly identified student performance based on the positive results. The true-negative rate and the false-negative rate are mathematically expressed as T_{NR} and F_{NR} ,

$$T_{NR} = \frac{T_N}{T_N + F_P} \quad (6)$$

$$F_{NR} = \frac{F_N}{F_N + T_P} \quad (7)$$

A comparative graph of this evaluation of the true-negative rate and the false-negative rate is plotted in Figure 7. As per the computed results, the AISAR system shows an increase in the true-negative rate with respect to the existing work. The negative rate denotes the rejection of correct or incorrect student performance. The increase in the true-negative rate is due to the correct prediction of a student's performance. The reduction in the false rates will increase the positive rate of the system. In previous work, student assessment was performed using decision tree algorithms that were able to categorise students but not evaluate individual performance. Now, in the proposed AISAR system, the score is predicted, then clustered, and then, from the cluster results, the individual performance is predicted using TMR. DBSCAN clustering is capable of supporting highly dense datasets. In contrast, the J48 decision tree-based algorithm cannot support larger datasets due to the increase in the depth of the tree.

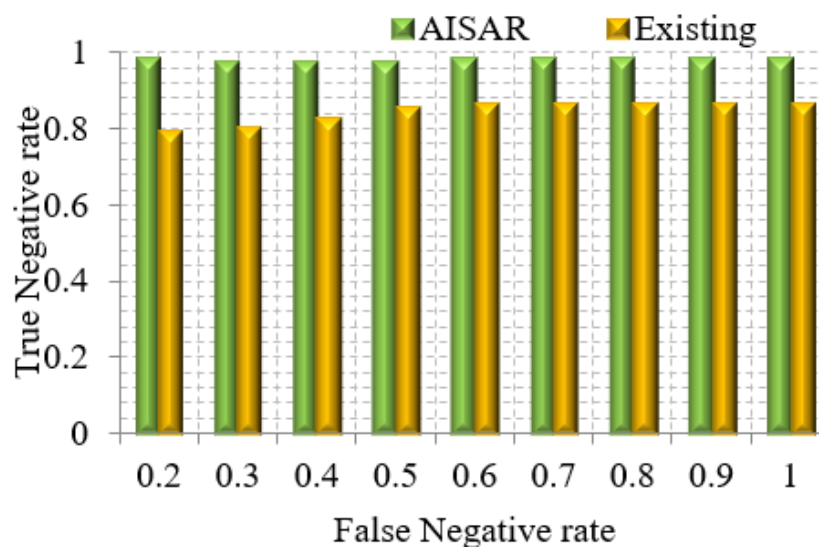


Figure 7. Comparison of false-negative rate versus true-negative rate.

The true-negative rate is 0.13 less in the existing work, but the increase in the dataset size will extend the difference value between the existing and AISAR systems. The proposed AISAR system attains a true-negative rate of 0.97; in contrast, the existing work reaches a true-negative rate of 0.84. The lower true-negative rate means that the system poorly predicts student performance. Hence, the proposed AISAR system is better than the existing machine learning algorithm.

5.2.3. Precision and Recall

Precision and recall are the key metrics that were estimated to evaluate the system's effectiveness in predicting student performance. Precision was measured as the ratio of properly determined positive results with respect to the total number of predicted correct results. Precision P_C was calculated as follows:

$$P_C = \frac{T_P}{T_P + F_P} \quad (8)$$

Recall was estimated from the accurately predicted results concerning the actual results. Recall was computed as follows:

$$R_C = \frac{T_P}{T_P + F_N} \quad (9)$$

The precision–recall plot is shown in Figure 8, where the precision is higher in the AISAR system than in the current work. This increase is due to an accurate prediction of student performance. In the AISAR system, student performance is predicted exactly

from the clusters using TMB, whereas machine learning algorithms fail to predict the performance of individual students. Hence, the AISAR system is better than the existing machine learning algorithms. The AISAR system reaches 0.91 precision compared to precision of only 0.606, which was achieved by the machine learning algorithms.

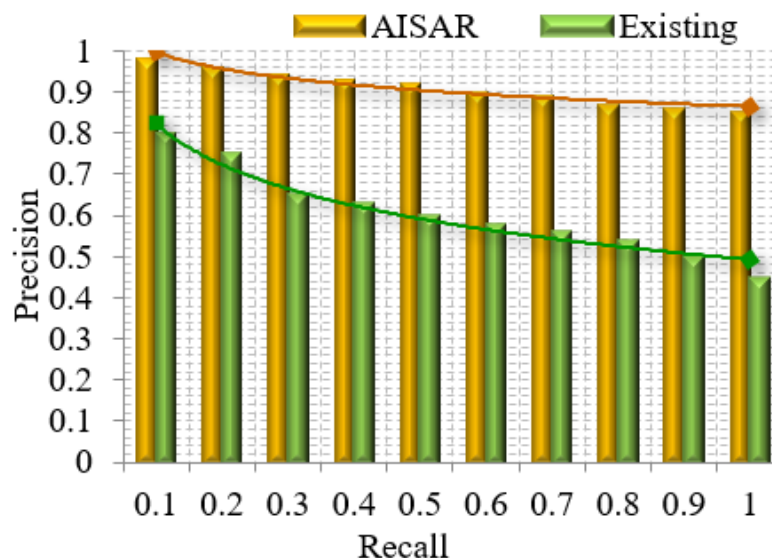


Figure 8. Comparison of precision and recall.

5.2.4. Accuracy

Accuracy is a key metric that was computed to illustrate the performance of the proposed system. This metric must increase to show that the system is operating very well by producing more accurate results. In the previous discussion, the proposed system performed better in the true-positive, true-negative, and precision results. These better performances reflect the improving accuracy of the system.

Table 6 illustrates the accuracy results for the machine learning algorithms and the AISAR system, which uses clustering and MapReduce to predict student performance. In contrast, the J48 decision tree algorithm has higher accuracy than the other machine learning algorithms, i.e., 88.52%. However, the proposed AISAR system achieves an accuracy of 97.21% due to the prediction of student performance by clustering following the MapReduce process. As a result, the increase in accuracy illustrates the better efficiency of the proposed AISAR system.

Table 6. Accuracy comparison.

Method	Accuracy (%)
Decision tree	85.91
J48	88.52
Classification and regression tree	82.25
Gradient boosting tree	86.43
Naïve Bayes	82.93
AISAR system	97.21

6. Conclusions

An e-learning system, known as the AISAR system, was developed in this work. This system focuses on student performance prediction and recommendations. The OULAD dataset was processed to test this system. Initially, the student score value was estimated using an RNN that considers student engagement information and test score information. From the computed score values, the students were grouped into clusters by the DBSCAN algorithm, which is capable of supporting clustering at an increased student density. The

purity of the DBSCAN clusters was high, i.e., the clusters were constructed without any mismatches in student score values. Then, the students with poor and average scores were processed in TMR to evaluate their performance. The threshold-based reducer process enabled us to predict the performance of individual students. Lastly, recommendations were provided in R-SARSA based on the student's performance. The rules were defined by the mean value of the student's score and engagement. In addition, to prevent unauthorised student access, the students were authenticated using their identity, password, and PIN. Hence, each student registered in the e-learning system was given the most appropriate recommendations.

Author Contributions: W.B.: system development; writing of the literature review; testing and analysing the data; writing and editing of the paper; and formatting. N.C.: supervised and managed the research; contributed and edited the literature review; and edited the paper. P.V.: co-supervised the research; contributed to the literature review; and edited the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This work was tested using the OULAD dataset. It is available online at https://analyse.kmi.open.ac.uk/open_dataset (accessed on 17 March 2020).

Acknowledgments: This work was supported by the Department of Computer Science and Information Technology at La Trobe University, Melbourne, Australia.

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

References

1. Tawafak, R.M.; Romli, A.B.; Arshah, R.B.A. E-learning Model for Students' Satisfaction in Higher Education Universities. In Proceedings of the International Conference on Fourth Industrial Revolution (ICFIR), Manama, Bahrain, 19–21 February 2019; pp. 1–6.
2. Andersson, C.; Kroisandt, G. Opportunities and Challenges with e-Learning Courses in Statistics for Engineering and Computer Science Students. In Proceedings of the IEEE World Engineering Education Conference (EDUNINE), Buenos Aires, Argentina, 11–14 March 2018; pp. 1–4.
3. Al-Rahmi, W.M.; Yahaya, N.; Aldraiweesh, A.A.; Alamri, M.M.; Aljarboa, N.A.; Alturki, U.; Aljeraiwi, A.A. Integrating Technology Acceptance Model with Innovation Diffusion Theory: An Empirical Investigation on Students' Intention to Use E-Learning Systems. *IEEE Access* **2019**, *7*, 26797–26809. [\[CrossRef\]](#)
4. Al-Rahmi, W.M.; Alias, N.; Othman, M.S.; Alzahrani, A.I.; Alfarraj, O.; Saged, A.A.; Rahman, N.S.A. Use of E-Learning by University Students in Malaysian Higher Educational Institutions: A Case in Universiti Teknologi Malaysia. *IEEE Access* **2018**, *6*, 14268–14276. [\[CrossRef\]](#)
5. Raspopovic, M.; Jankulovic, A. Performance Measurement of E-Learning Using Student Satisfaction Analysis. *Inf. Syst. Front.* **2017**, *19*, 869–880. [\[CrossRef\]](#)
6. McConnell, D. E-learning in Chinese Higher Education: The View from Inside. *High. Educ.* **2018**, *75*, 1031–1045. [\[CrossRef\]](#)
7. Kew, S.N.; Petsangsri, S.; Ratanaolarn, T.; Tasir, Z. Examining the Motivation Level of Students in E-Learning in Higher Education Institution in Thailand: A Case Study. *Educ. Inf. Technol.* **2018**, *23*, 2947–2967. [\[CrossRef\]](#)
8. Ashwin, T.S.; Guddeti, R.M.R. Impact of Inquiry Interventions on Students in E-Learning and Classroom Environments Using Affective Computing Framework. *User Modeling User-Adapt. Interact.* **2020**, *30*, 759–801. [\[CrossRef\]](#)
9. Fatahi, S. An Experimental Study on an Adaptive E-Learning Environment based on Learner's Personality and Emotion. *Educ. Inf. Technol.* **2019**, *24*, 2225–2241. [\[CrossRef\]](#)
10. Gao, H.; Wu, H.; Wu, X. Chances and Challenges: What E-learning Brings to Traditional Teaching. In Proceedings of the 2018 9th International Conference on Information Technology in Medicine and Education (ITME), Hangzhou, China, 19–21 October 2018; pp. 420–422.
11. El Mhouthi, A.; Erradi, M.; Nasseh, A. Using Cloud Computing Services in E-Learning Process: Benefits and Challenges. *Educ. Inf. Technol.* **2018**, *23*, 893–909. [\[CrossRef\]](#)
12. Vershitskaya, E.R.; Mikhaylova, A.V.; Gilmanishina, S.I.; Dorozhkin, E.M.; Epaneshnikov, V.V. Present-Day Management of Universities in Russia: Prospects and Challenges of E-learning. *Educ. Inf. Technol.* **2020**, *25*, 611–621. [\[CrossRef\]](#)
13. Souabi, S.; Retbi, A.; Idrissi, M.K.I.K.; Bennani, S. Recommendation Systems on E-Learning and Social Learning: A Systematic Review. *Electron. J. e-Learn.* **2021**, *19*, 432–451. [\[CrossRef\]](#)

14. Li, H.H.; Liao, Y.H.; Wu, Y.T. Artificial Intelligence to Assist E-Learning. In Proceedings of the 14th International Conference on Computer Science & Education (ICCSE), Toronto, ON, Canada, 19–21 August 2019; pp. 653–654.
15. Chanaa, A. Deep Learning for a Smart E-Learning System. In Proceedings of the 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech), Brussels, Belgium, 26–28 November 2018; pp. 1–8.
16. Fok, W.W.; He, Y.S.; Yeung, H.A.; Law, K.Y.; Cheung, K.H.; Ai, Y.Y.; Ho, P. Prediction Model for Students' Future Development by Deep Learning and TensorFlow Artificial Intelligence Engine. In Proceedings of the 4th International Conference on Information Management (ICIM), Oxford, UK, 25–27 May 2018; pp. 103–106.
17. Khanal, S.S.; Prasad, P.W.C.; Alsadoon, A.; Maag, A. A Systematic Review: Machine Learning Based Recommendation Systems for E-learning. *Educ. Inf. Technol.* **2020**, *25*, 2635–2664. [\[CrossRef\]](#)
18. Mansur, A.B.F.; Yusof, N.; Basori, A.H. Personalized Learning Model Based on Deep Learning Algorithm for Student Behaviour Analytic. *Procedia Comput. Sci.* **2019**, *163*, 125–133. [\[CrossRef\]](#)
19. Zhang, K.; Aslan, A.B. AI Technologies for Education: Recent Research & Future Directions. *Comput. Educ. Artif. Intell.* **2021**, *2*, 100025. [\[CrossRef\]](#)
20. Sousa, M.; Dal Mas, F.; Pesqueira, A.; Lemos, C.; Verde, J.M.; Cobianchi, L. The Potential of AI in Health Higher Education to Increase the Students' Learning Outcomes. *TEM J.* **2021**, *2*, 488–497. [\[CrossRef\]](#)
21. Tan, J. Information Analysis of Advanced Mathematics Education-Adaptive Algorithm Based on Big Data. *Math. Probl. Eng.* **2022**, *2022*, 7796681. [\[CrossRef\]](#)
22. Wan, S.; Niu, Z. A Hybrid E-Learning Recommendation Approach Based on Learners' Influence Propagation. *IEEE Trans. Knowl. Data Eng.* **2019**, *32*, 827–840. [\[CrossRef\]](#)
23. Rahman, M.M.; Abdullah, N.A. A Personalized Group-Based Recommendation Approach for Web Search in E-learning. *IEEE Access* **2018**, *6*, 34166–34178. [\[CrossRef\]](#)
24. Lai, R.; Wang, T.; Chen, Y. Improved Hybrid Recommendation with User Similarity for Adult Learners. *J. Eng.* **2019**, *11*, 8193–8197. [\[CrossRef\]](#)
25. Hassan, M.A.; Habiba, U.; Khalid, H.; Shoaib, M.; Arshad, S. An Adaptive Feedback System to Improve Student Performance Based on Collaborative Behavior. *IEEE Access* **2019**, *7*, 107171–107178. [\[CrossRef\]](#)
26. Yağcı, M. Educational Data Mining: Prediction of Students' Academic Performance Using Machine Learning Algorithms. *Smart Learn. Environ.* **2022**, *9*, 1–19. [\[CrossRef\]](#)
27. Redondo, J.M. Improving Student Assessment of a Server Administration Course Promoting Flexibility and Competitiveness. *IEEE Trans. Educ.* **2018**, *62*, 19–26. [\[CrossRef\]](#)
28. Balderas, A.; De-La-Fuente-Valentin, L.; Ortega-Gomez, M.; Dodero, J.M.; Burgos, D. Learning Management Systems Activity Records for Students' Assessment of Generic Skills. *IEEE Access* **2018**, *6*, 15958–15968. [\[CrossRef\]](#)
29. Akram, A.; Fu, C.; Li, Y.; Javed, M.Y.; Lin, R.; Jiang, Y.; Tang, Y. Predicting students' academic procrastination in blended learning course using homework submission data. *IEEE Access* **2019**, *7*, 102487–102498. [\[CrossRef\]](#)
30. Sera, L.; McPherson, M.L. Effect of A Study Skills Course on Student Self-Assessment of Learning Skills and Strategies. *Curr. Pharm. Teach. Learn.* **2019**, *11*, 664–668. [\[CrossRef\]](#)
31. Cerezo, R.; Bogarín, A.; Esteban, M.; Romero, C. Process Mining for Self-Regulated Learning Assessment in E-learning. *J. Comput. High. Educ.* **2020**, *32*, 74–88. [\[CrossRef\]](#)
32. Garg, R.; Kumar, R.; Garg, S. MADM-Based Parametric Selection and Ranking of E-Learning Websites Using Fuzzy COPRAS. *IEEE Trans. Educ.* **2019**, *62*, 11–18. [\[CrossRef\]](#)
33. Al-Tarabily, M.M.; Abdel-Kader, R.F.; Azeem, G.A.; Marie, M.I. Optimizing Dynamic Multi-Agent Performance in E-learning Environment. *IEEE Access* **2018**, *6*, 35631–35645. [\[CrossRef\]](#)
34. Govindasamy, K.; Velmurugan, T. Analysis of Student Academic Performance Using Clustering Techniques. *Int. J. Pure Appl. Math.* **2018**, *119*, 309–323.
35. Kausar, S.; Huahu, X.; Hussain, I.; Wenhao, Z.; Zahid, M. Integration of Data Mining Clustering Approach in The Personalized E-Learning System. *IEEE Access* **2018**, *6*, 72724–72734. [\[CrossRef\]](#)
36. Marras, M.; Boratto, L.; Ramos, G.; Fenu, G. Equality of Learning Opportunity Via Individual Fairness in Personalized Recommendations. *Int. J. Artif. Intell. Educ.* **2021**, 1–49. [\[CrossRef\]](#)
37. Elkhateeb, M.; Shehab, A.; El-Bakry, H. Mobile Learning System for Egyptian Higher Education Using Agile-Based Approach. *Educ. Res. Int.* **2019**, *2019*, 7531980. [\[CrossRef\]](#)
38. Wan, S.; Niu, Z. An E-Learning Recommendation Approach Based on The Self-Organization of Learning Resource. *Knowl. Based Syst.* **2018**, *160*, 71–87. [\[CrossRef\]](#)
39. Gulzar, Z.; Leema, A.A.; Deepak, G. Pcrs: Personalized Course Recommender System Based on Hybrid Approach. *Procedia Comput. Sci.* **2018**, *125*, 518–524. [\[CrossRef\]](#)
40. Dahdouh, K.; Dakkak, A.; Oughdir, L.; Ibriz, A. Large-Scale E-Learning Recommender System Based on Spark and Hadoop. *J. Big Data* **2019**, *6*, 1–23. [\[CrossRef\]](#)
41. He, H.; Zhu, Z.; Guo, Q.; Huang, X. A Personalized E-learning Services Recommendation Algorithm Based on User Learning Ability. In Proceedings of the 2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT), Maceio, Brazil, 5–18 July 2019; Volume 2161, pp. 318–320.

42. Samin, H.; Azim, T. Knowledge Based Recommender System for Academia Using Machine Learning: A Case Study on Higher Education Landscape of Pakistan. *IEEE Access* **2019**, *7*, 67081–67093. [[CrossRef](#)]
43. Okada, A.; Whitelock, D.; Holmes, W.; Edwards, C. e-Authentication for Online Assessment: A Mixed-Method Study. *Br. J. Educ. Technol.* **2019**, *50*, 861–875. [[CrossRef](#)]
44. Smith, A.; Leeman-Munk, S.; Shelton, A.; Mott, B.; Wiebe, E.; Lester, J. A Multimodal Assessment Framework for Integrating Student Writing and Drawing in Elementary Science Learning. *IEEE Trans. Learn. Technol.* **2019**, *12*, 3–15. [[CrossRef](#)]
45. Hussain, M.; Zhu, W.; Zhang, W.; Abidi, S.M.R. Student Engagement Predictions in an E-Learning System and Their Impact on Student Course Assessment Scores. *Comput. Intell. Neurosci.* **2018**, *2018*, 6347186. [[CrossRef](#)]
46. Brahim, B.; Lotfi, A. A Traces-Based System Helping to Assess Knowledge Level in E-Learning System. *J. King Saud Univ. Comput. Inf. Sci.* **2020**, *32*, 977–986. [[CrossRef](#)]
47. Lee, A.; Han, J.Y. Effective User Authentication System in an E-Learning Platform. *Int. J. Innov. Creat. Change* **2020**, *13*, 1101–1113.
48. Rodríguez-Hernández, C.F.; Musso, M.; Kyndt, E.; Cascallar, E. Artificial Neural Net-works in Academic Performance Prediction: Systematic Implementation and Predictor Evaluation. *Comput. Educ. Artif. Intell.* **2021**, *2*, 100018. [[CrossRef](#)]
49. Zhang, Q.; Lu, J.; Jin, Y. Artificial Intelligence in Recommender Systems. *Complex Intell. Syst.* **2021**, *7*, 439–457. [[CrossRef](#)]
50. Zhai, X.; Chu, X.; Chai, C.S.; Jong, M.S.Y.; Istenic, A.; Spector, M.; Liu, J.B.; Yuan, J.; Li, Y. A Review of Artificial Intelligence (AI) in Education from 2010 to 2020. *Complexity* **2021**, *2021*, 8812542. [[CrossRef](#)]
51. Kuzilek, J.; Hlosta, M.; Zdrahal, Z. Open University Learning Analytics Dataset. *Sci. Data* **2017**, *4*, 1–8. [[CrossRef](#)]
52. Siddique, S.A. Improvement of Online Course Content Using MapReduce Big Data Analytics. *Int. Res. J. Eng. Technol.* **2020**, *07*, 50–56.