

Article

A Case-Study Comparison of Machine Learning Approaches for Predicting Student's Dropout from Multiple Online Educational Entities

José Manuel Porras, Juan Alfonso Lara , Cristóbal Romero * and Sebastián Ventura 

Department of Computer Science, University of Cordoba, 14071 Córdoba, Spain; i82poroj@uco.es (J.M.P.); juan.lara@uco.es (J.A.L.); sventura@uco.es (S.V.)

* Correspondence: cromero@uco.es

Abstract: Predicting student dropout is a crucial task in online education. Traditionally, each educational entity (institution, university, faculty, department, etc.) creates and uses its own prediction model starting from its own data. However, that approach is not always feasible or advisable and may depend on the availability of data, local infrastructure, and resources. In those cases, there are various machine learning approaches for sharing data and/or models between educational entities, using a classical centralized machine learning approach or other more advanced approaches such as transfer learning or federated learning. In this paper, we used data from three different LMS Moodle servers representing homogeneous different-sized educational entities. We tested the performance of the different machine learning approaches for the problem of predicting student dropout with multiple educational entities involved. We used a deep learning algorithm as a predictive classifier method. Our preliminary findings provide useful information on the benefits and drawbacks of each approach, as well as suggestions for enhancing performance when there are multiple institutions. In our case, repurposed transfer learning, stacked transfer learning, and centralized approaches produced similar or better results than the locally trained models for most of the entities.

Keywords: dropout prediction; predictive analytics; transfer learning; federated learning



Citation: Porras, J.M.; Lara, J.A.; Romero, C.; Ventura, S. A Case-Study Comparison of Machine Learning Approaches for Predicting Student's Dropout from Multiple Online Educational Entities. *Algorithms* **2023**, *16*, 554. <https://doi.org/10.3390/a16120554>

Academic Editor: Ioannis Tsoulos

Received: 7 November 2023

Revised: 1 December 2023

Accepted: 1 December 2023

Published: 3 December 2023



Copyright: © 2023 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

One important task in learning analytics (LA) [1] and educational data mining (EDM) [2] research is student dropout prediction (SDP). SDP is an important educational problem because of the high dropout rate, mainly from e-learning environments. The recent spread of online courses—with enormous numbers of enrolled students, only a fraction of whom complete their studies successfully—has led to increased attention to this problem. As a consequence, there is a growing interest in the adoption of automated systems for predicting student dropout. Automated methodologies have also caught the attention of researchers, particularly in the area of machine learning. Students doing online degree programs have a higher chance of dropping out than those attending conventional classroom environments. According to [3], 40–80% of online students drop out from online classes. Moreover, students may leave courses at any time without notice or further repercussions. Therefore, it is of paramount interest to find more effective methods of addressing the problem of dropout in e-learning environments. The specific objective of SDP in e-learning environments is to model student behavior interacting with e-learning platforms to determine which students are likely to not finish a course, degree, etc. [4].

Predicting student dropout in online learning is a challenge that has been widely studied in the literature [5]. Educational data mining (EDM) techniques have been successfully applied [2] to solve this problem as a binary classification task (zero or one). In this task, there is a set of already labeled training data (students who have already completed the course). Therefore, it is known whether they dropped out (labeled class 1)

or successfully completed the course (labeled class 0). The task aims to make the earliest possible prediction of the label or class for new students taking a new course. Traditionally, a wide range of classical machine learning and data mining algorithms have been used to solve this classification problem [6–8] such as decision trees, Bayesian networks, support vector machines, etc. But in recent years, new, more powerful classification algorithms have appeared, such as deep learning algorithms. They have significantly better predictive capabilities than classical algorithms [9,10]. In this study, we used a deep learning classifier as an algorithm for predicting student dropout.

At the outset, it is important to define a term that we use throughout this paper. The term “educational entity” generally refers to any organization or institution that provides educational services, such as a school, college, university, or any other type of learning center. Traditionally, each educational entity builds its own individual predictive model from its own data. These models are commonly referred to as individual or local models. However, this method requires each entity to collect enough data to train its individual model, which takes time and would require resources to create and maintain the prediction model. Furthermore, an entity may be subdivided into smaller entities, for example, a university with its individual faculties, departments, etc., or a group of universities in the same country. In that case, it is crucial to have the necessary data and resources to build and manage the model for each of those subentities and the costs that they entail. There may be subentities with few resources or that do not have the data available to develop their own prediction models.

When multiple entities share their data, a more diverse and representative dataset is created, which can result in more robust and more broadly applicable models. However, data sharing also poses significant privacy and security challenges, so any collaborative effort involving data sharing must be conducted with a high level of confidentiality and in compliance with privacy regulations [11,12]. Conversely, sharing machine learning models between educational entities offers unique benefits and drawbacks. By sharing models, entities can benefit from the knowledge of models trained elsewhere. This can expedite the development of effective models, and shared models can serve as strong starting points for modification to specific local contexts. Nevertheless, models may contain implicit information about the original data and may even be susceptible to reverse engineering attacks in certain situations. Therefore, it is critical to ensure that shared models do not compromise data privacy or reveal personal information. Additionally, adapting shared models to local data may require significant resources, and it is important to keep in mind that a model that performs well in one context may not always be immediately applicable in another. There are different approaches where entities can use models already trained elsewhere or share data with other entities to create more reliable and generalizable models for later use. Each approach has its advantages and disadvantages, such as data size, data variability, privacy concerns, and generalization issues.

The most common data sharing approach consists of using a centralized learning approach, where data from different sources are gathered together, and then, predictive models are built to find common patterns among all of them. This approach has been widely used by many authors in order to have a one-size-fits-all general model. Some authors who started to consider it as a particular approach for data combination for different sources in education include [13,14]. However, as those authors stated, the centralized approach has some limitations due to legal and privacy issues, and therefore it is quite often not possible to combine data.

Over recent years, several transfer learning studies have been performed in the field of EDM, where models are built in a context (a course, for instance) and transferred to be used in another context. There is a wide variety, as they include very trivial approaches and other, more complex ones. A large group of studies have used deep learning techniques to build models that are subsequently transferred. In [15], the authors suggested that a model trained in one course can be replicated for use in other courses, so that the weights of the target course are initialized with the weights learned in the source course, and from

there training is continued towards the target. There have been innovative studies with advanced techniques that use ensembles, which are strategies for combining decisions taken by several machine learning models to improve generalization. This is the proposal in [16], where local training is first performed on the nodes and then two ensemble transfer strategies are used: the first, based on voting (the decision is made based upon the votes from each node); and the second, based on stacking—an approach in which the decisions adopted by the different models feed a new model. In addition, many studies have used a more trivial approach consisting of direct transfer of a model trained on one source for immediate use on the target [17–21].

When attempting to combine information in that context, there are situations where the data cannot be merged directly. For example, when entities are not authorized to share data due to legal privacy issues [22]. This has led to the emergence of so-called federated learning (FL) methods, which allow multiple entities to build a common machine learning model without sharing data [23]. Most FL methods use the idea of averaging the values provided by the sources for each model parameter, with education being a domain where this idea is also widespread [24,25]. In [14], average values were calculated of the weights of the neural networks provided by the participating nodes for the prediction of student dropout to demonstrate that this federated approach improved on the results of a centralized approach and avoided the concentration of the data, thus maintaining privacy.

Several studies have proposed some of the above approaches, mainly transfer learning, to address the problem of student dropout prediction between different instances of the same course [26] or from one course to a different, related or unrelated course [15,27], while others addressed the problem with different universities involved [16]. The novelty of the current paper is specifically that it evaluates and compares the performance of all the previously proposed machine learning approaches when there are several, different-sized entities involved.

In this paper, we explore all the previously mentioned approaches (individual, centralized, transfer learning, and federated learning) applied to the SDP problem. To our knowledge, this is the first study to compare all these approaches together. We present experimental results that highlight the strengths and limitations of each approach based on the results from applying the aforementioned collaborative approaches to datasets from three different-sized educational entities. Ultimately, our goal is to provide an initial understanding of how all these approaches can influence the creation of accurate predictive models in real-world situations with several different-sized entities. The main contribution of this paper is the proposal of a series of findings to guide practitioners and the research community toward informed decision-making when addressing predictive modeling from different entities.

The rest of this paper is organized as follows: In Section 2, we examine the main approaches for generating prediction models starting from different entities. In Section 3, we present the materials and methods, including the datasets used and the neural network architecture we used. In Section 4, we present the results of our experiment findings and in Section 5, we discuss these results and explore practical implications arising from our discoveries. Finally, in Section 6, we conclude the paper by presenting our findings and discussing their significance for both EDM as a field and for student dropout prediction. Furthermore, we propose directions for future research that could further enhance our understanding in this domain.

2. Machine Learning Approaches

There are various machine learning strategies or approaches that can be used to create prediction models of student dropout starting from multiple educational entities [13,22]. The different approaches can be compared in terms of sharing data and collaboration to generate models. Some authors [16] use the term multi-institutional when using data from multiple institutions or entities during training and cross-institutional when using models from multiple institutions or entities for inference. These strategies vary from

traditional individual models to centralized models to more updated transfer and federated learning models. A summary of the main learning approaches is shown in Figure 1. In the individual approach, each entity maintains its data and models independently, which limits collaboration and joint use of resources. In contrast, the centralized approach relies on the consolidation of data from different entities into a single dataset. Meanwhile, transfer learning allows the reuse of pre-trained models from one entity to another. Finally, federated learning is based on the consolidation of a central model based on models trained at each entity separately.

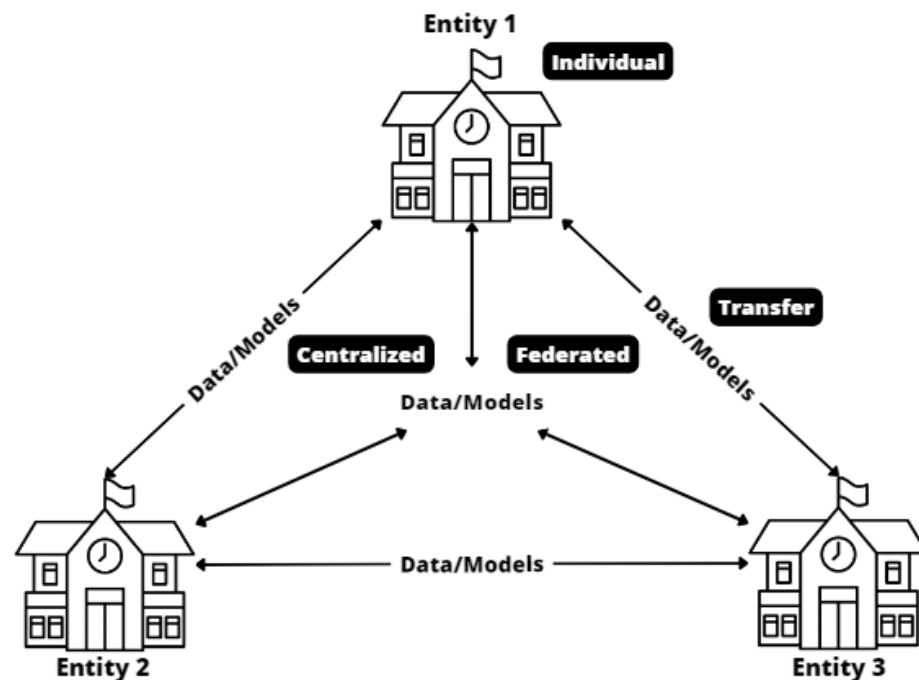


Figure 1. Summary of main approaches for generating prediction models starting from different entities.

2.1. Individual Models

The use of local or individual models is the traditional approach where each entity uses its own specific prediction model trained from its own data [13]. One of the potential benefits lies in the ability to adapt to the unique peculiarities of that data. This can result in robust performance when predicting student dropout in that particular dataset. However, these models have additional challenges, such as the time needed to collect enough data or duplication of effort and resources needed to develop specific entity models, as well as the difficulty in applying the findings in a broader context.

The process (see Figure 2) begins with data collection from each entity (1). With the data prepared, the machine learning model is then designed and trained (2). Finally, once the model has been trained and validated, it is deployed in its own entity where it is used to make predictions and decisions based on each entity's unique data (3).

2.2. Centralized Model

Centralized learning [13] is a data analysis strategy that involves training a single model using combined data from multiple entities. The centralized model optimizes performance by leveraging the unique characteristics of each dataset, leading to the creation of models that are more generalizable. However, it is crucial to understand that there may be issues such as noise, bias, and even privacy concerns when using data that are shared between entities.

In this case (see Figure 3), each entity (1) sends its raw data to the central server (2). Subsequently, the central server combines all the data from all the different entities' students and then trains a machine learning model using the combined data (3). As a result, a single integrated model is created, which is used to predict dropout (4).

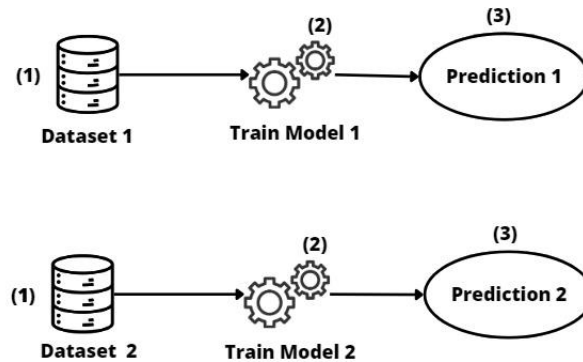


Figure 2. Individual models approach.

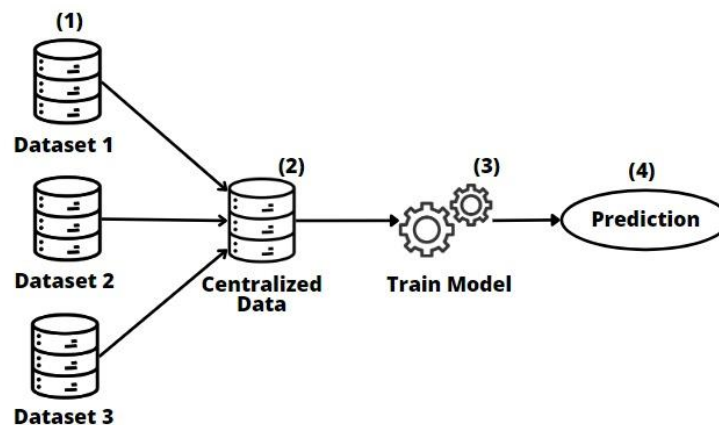


Figure 3. Centralized model approach.

2.3. Transfer Learning

Transfer learning follows the idea of using the learning in a model trained on one specific source in another specific context. There are different versions of transfer learning such as direct transfer learning, repurposed transfer learning, in which the model retrains on the target data, and ensemble-based transfer learning models such as soft voting and stacking transfer learning.

2.3.1. Direct Transfer

Direct transfer learning [16] is used to leverage knowledge and insights gained from one dataset or entity to improve performance in another related domain or dataset. In this strategy, models can benefit from patterns and representations learned from the source data and apply them directly to the target data. However, the performance of direct transfer may largely depend on the match between the sizes of each entity.

The process is described in Figure 4. Firstly, the information is collected from the source entity, which acts as a source of knowledge. (1) These data are used to train an initial model that captures fundamental patterns and features (2). Then, these models are directly transferred or applied to the target data of interest to make predictions (3).

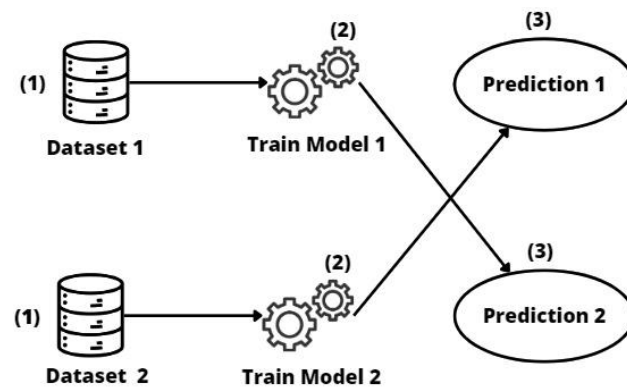


Figure 4. Direct transfer learning approach.

2.3.2. Repurposed Transfer

Repurposed transfer learning [15] is a machine learning approach that leverages knowledge learned from a pretrained model to retrain it to be applied effectively to related but different tasks with different datasets. The key idea is to adapt and reuse the knowledge learned by a pretrained model, fine-tuning its parameters on a task-specific dataset (retrain phase). It is important to underscore that the implementation of repurposed models may require additional resources for adaptation to specific local contexts.

This approach (see Figure 5), like the previous one, starts with data collection in entities or relevant datasets, which act as a source of knowledge (1). These data are used to train an initial model that captures fundamental patterns and features (2). Then, these models are transferred to the target data, and additional training is carried out on the target data. This stage involves fine-tuning or adjusting the pretrained model to adapt it to the peculiarities of the new target data.

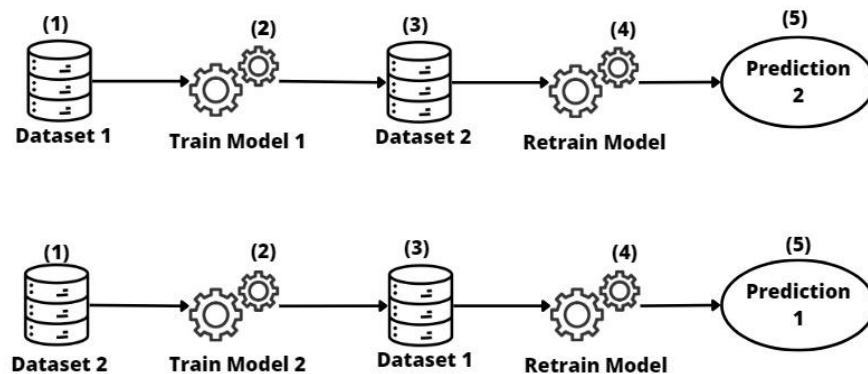


Figure 5. Repurposed Transfer Learning approach.

2.3.3. Soft-Voting Transfer

Soft-voting transfer [16] is an ensemble approach that uses an averaging mechanism to combine the results of models “voters” trained on separate entities. The soft-voting approach tries to improve collaboration and efficiency in early dropout detection without having to create extra models.

The process (see Figure 6) begins with collecting data from relevant distributions or entities (1). Each model or “voter” is trained on one of these separate distributions, with no training at the target entity in a “zero-shot” setup (2). Then, during the prediction stage, a soft-voting mechanism is used to combine the probabilities predicted by each model (3–4). Unlike hard voting, where decisions are made based on majority votes, soft voting averages the predicted probabilities of each model.

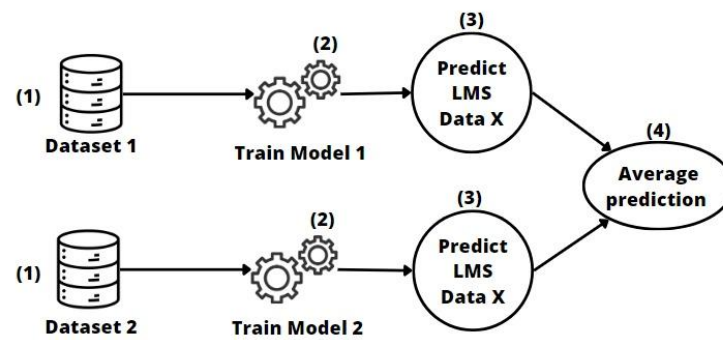


Figure 6. Soft-voting transfer learning approach.

2.3.4. Stacking Transfer

Another ensemble approach is stacking transfer which uses stacked generalization [16] to combine the predictions of models trained on all available entities with the training data of the source entity. This is achieved by concatenating the predictions of each classifier to the input features and creating a new classifier from this concatenated data. This approach is especially valuable when working with multiple entities or data sources, as it allows predictions from models trained on different sources to be combined to improve generalization and performance on a specific target task. However, it is important to note that implementing this technique may require more computational resources and effort than other approaches.

In summary, training with stacking (see Figure 7) involves concatenating predictions (2) from models trained on different entities (1) with the input features (3). Then, a new classifier is trained (4) using this new dataset, allowing knowledge from various sources to be leveraged to improve model performance on a specific task (5).

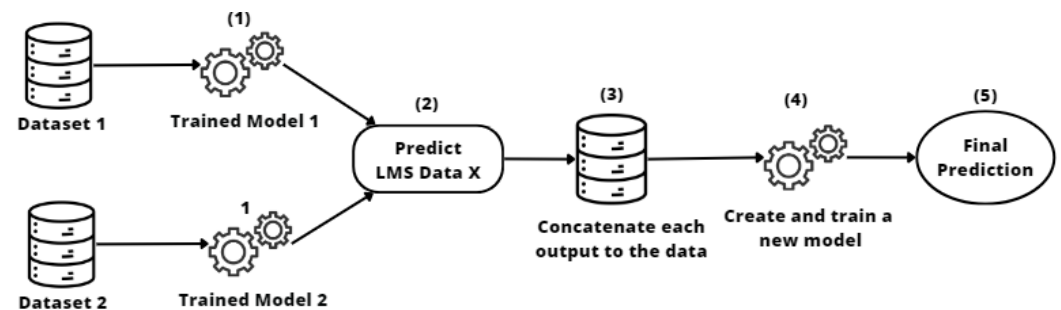


Figure 7. Stacking transfer learning approach.

2.4. Federated Learning

Federated Learning addresses privacy and security issues using collaborative training without sharing data [23]. This approach typically comprises two variants: horizontal (datasets share the same feature space but differ in samples) and vertical (datasets share the same sample identifier but differ in feature space). In our study, we used the horizontal variant of federated learning, where the data structure across different entities remains consistent.

In this approach (see Figure 8), each entity has its own dataset (1), and to compute the final model, an iterative process is used for model parameter estimation within the clients and the server. In each iteration, specific clients are chosen to train the model with their own data locally (2). Then, the server aggregates all clients' results to update the model state (3 and 4) and update the final global model (5) using an average of the local parameter estimations.

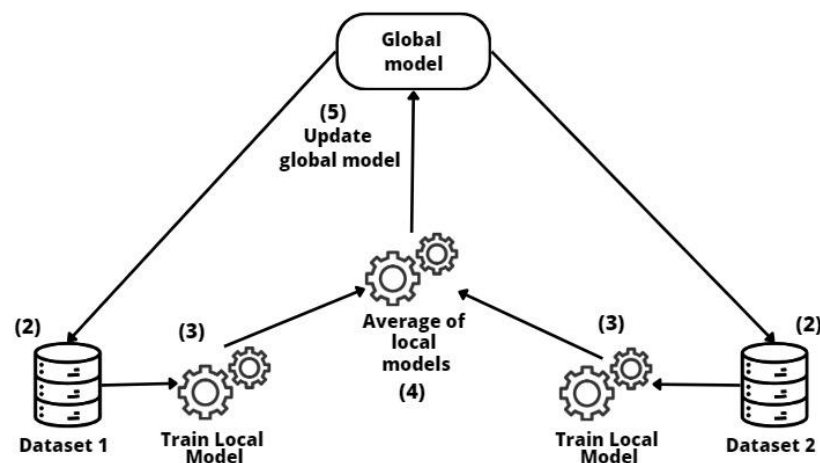


Figure 8. Federated learning approach.

3. Materials and Methods

The objective of our experiments, as shown in Figure 9, was to compare and evaluate different machine learning approaches (described in Section 2 above) for student dropout detection when having multiple entities' datasets available. We started by collecting data from multiple educational entities (left of Figure 9). Then, we applied the proposed approaches from local/individual to centralized models, going from different versions of transfer learning to federated learning (central part of Figure 9). We used the same deep learning prediction model as the classifier to compare the different approaches. Finally, our study sought to identify the most effective approach, its advantages and drawbacks (right of Figure 9), and how performance varied when using different-sized datasets.

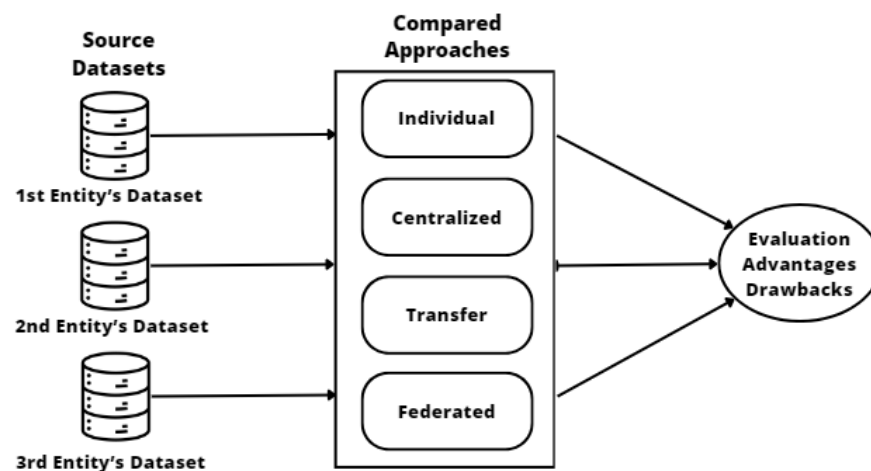


Figure 9. Experimental design scheme.

3.1. Data

All of the anonymized data in this study were provided by three online higher-education entities. The data came from student interaction data with Moodle courses. As Figure 10 shows, the student interaction data were sequentially grouped by week. It is also important to note that a student could start and drop out a course in any week during the course. Thus, for each student, there was a sequence of weeks or rows in the data file where their interaction activity with the course was stored. The length of the sequence (number of rows) for each learner was variable depending on the number of weeks where each learner presented interaction information. The final status of each student: dropout (value 1) or nondropout (value 0), was added as a class to predict. A dropout student in our context referred to a student who enrolled in an online course but did not complete it.

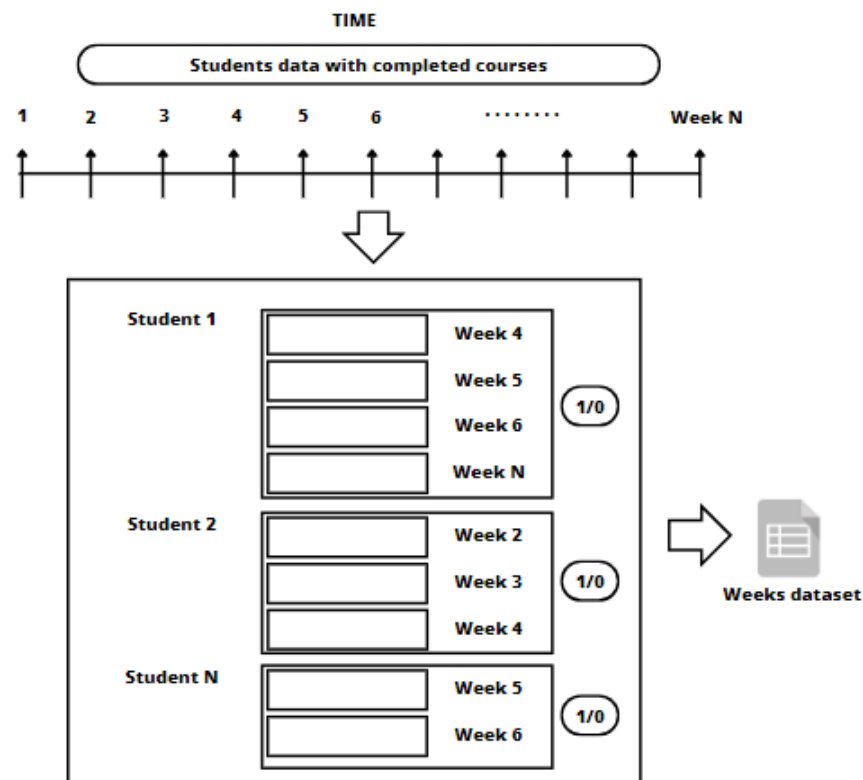


Figure 10. Data approach by weeks.

The data used in this paper came from three different LMS (learning management system) Moodle servers representing three educational entities with diverse students and courses (see Figure 11):

- The first entity's dataset comprised 224,136 students and 7706 courses of which 174,556 students completed their courses (78%), and 49,580 dropped out before finishing the course (22%). This was a large dataset.
- The second entity's dataset comprised 98,829 students and 5871 courses of which 74,048 students completed their courses (75%), and 24,781 dropped out before finishing the course (25%). This was a medium dataset.
- The third entity's dataset comprised 15,023 students and 517 courses of which 10,552 students completed their courses (77%), and 4382 dropped out before finishing the course (23%). This was a small dataset.

Figure 11 shows that the distribution of classes was unbalanced, but there was a similar percentage of dropout in the three entities' data (between 22% and 25%).

We used the same attributes to predict student dropout for the three entities. We used only attributes related to students' activity on their courses in the LMS Moodle system, avoiding any bias in the data and trying to achieve homogeneous entity datasets. The specific attributes are shown in Table 1. We only selected summarization attributes or features that could be directly obtained from students' interaction with the online courses. Our objective was to be able to generate prediction models that could be directly applied or transferred to any learning management systems (LMS), massive online open courses (MOOCs), etc. These summarization variables can be easily obtained from any type of online educational environment. In our case, the 11 specifically used variables were provided by the online course provider, and we only had to preprocess them.

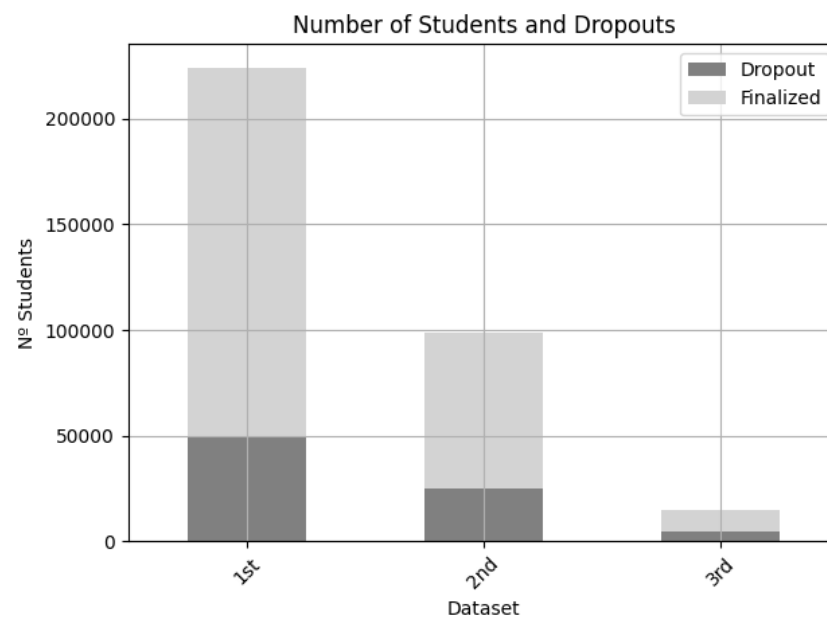


Figure 11. Students and label distribution in each entity's dataset.

Table 1. Description of student interaction data.

Variable Name	Description
discussions_viewed	Number of discussions the student has seen
htm_completed	Number of HTML pages completed
vid_completed	Number of completed video views
exa_completed	Number of examinations completed
autoeva_completed	Number of self-assessments completed
n_posts	Number of posts or messages viewed
n_discussions	Number of discussions held in the period
assignment_submitted	Number of tasks performed
com_tutor	Number of times connected to the tutor
Hours in course	Number of hours spent
Course visits	Number of visits to the course

The values of these attributes were preprocessed as follows:

- Data cleaning: if a null or unknown value was found in any attribute, it was replaced with a value of 0.
- Data normalization/scaling: to bring all numerical values into the same range, they were rescaled to the range [0–1] using the standard min–max scaler normalization, as shown in Equation (1):

$$X_{scaled} = \frac{(X - X_{min})}{(X_{max} - X_{min})}, \quad (1)$$

3.2. Prediction Model

The same deep learning classifier or prediction model was used for all the machine learning approaches in order to compare them. We selected a bidirectional long short-term memory (BiLSTM) neural network (Figure 12), which allowed us to capture dependencies and patterns in sequential data. We used this specific classifier because it gave the best results when comparing other classification models in previous work [28].

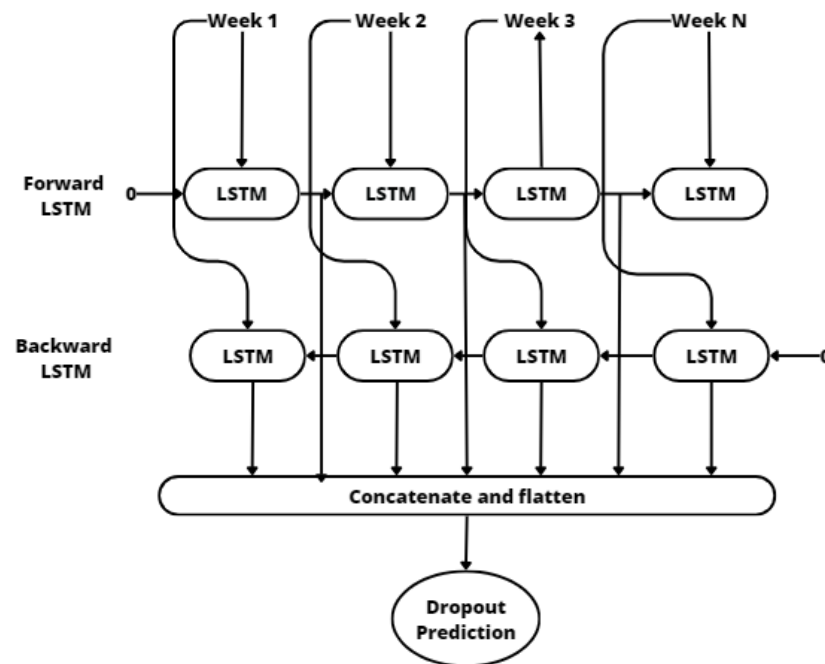


Figure 12. BiLSTM prediction model.

In this type of neural network, there is a bidirectional LSTM [29] layer that is responsible for learning and representing complex patterns and dependencies within the input sequence. As Figure 12 shows, it contains a variable number of LSTM units, which have internal memory cells to store and update information over time. These memory cells enable the network to capture long-term dependencies and retain important information throughout the sequence. In our research, the input data from the different weeks of the course (top side of Figure 12) were fed into a 128-neuron BiLSTM layer with 128 LSTM forward neurons and 128 LSTM backward neurons. The bidirectional nature of the BiLSTM enables the network to use information from both past and future contexts, enhancing its ability to understand the temporal dynamics of the data. The output of the BiLSTM layer is then passed to the final output layer. In our case, this output layer consisted of a single neuron with a sigmoid activation function. The sigmoid activation function ensures that the output is bounded between 0 and 1, representing a probability of our binary classification problem, 0 for nondropout, and 1 representing dropout.

Deep neural networks are represented by a number of connecting weights between the layers. During the training process, these weights are adjusted in order to minimize the error of the expected output. The parameters of the BiLSTM layer and the output layer were updated using the Adam optimizer, which adjusts the weights of the neurons based on the gradient of a loss function.

We used binary cross-entropy as the loss function, which measures the dissimilarity between the predicted outputs and the true target values for binary classification tasks. The BiLSTM algorithm was implemented in Python using the libraries: scikit-learn, tensorflow, and keras. We set class weights to solve the problem of class imbalance. More specifically, we set a higher weight to the minority class by using the `compute_class_weight` function from the scikit-learn library.

4. Results

This section describes the results from our experiments for evaluating the effectiveness of the different ML approaches using the BiLSTM neural network on the three educational datasets or entities presented in Section 3.1. Our main goal was to understand how performance varied for each approach and reveal their main advantages and drawbacks. We followed the same methodology for all experiments with the three entities' datasets

and the different approaches. We executed the deep learning model in a 10-fold cross-validation to ensure the reliability of our results through training and testing with different partitions of the datasets in each iteration. This allowed us to understand how well each method performed overall and its ability to handle new and unexpected data. As a baseline for comparison, we considered the individual or local model trained and tested at the same entity.

To evaluate the predictive performance of our models, taking into account the imbalance in the data and ensuring a robust assessment of effectiveness in predicting student dropout, we used the two following metrics:

1. AUC-ROC: The area under the ROC curve is one of the most important metrics used to represent the expected performance of a classifier based on the area under the receiver operating characteristic curve (AUC). The AUC is formally defined as the result of applying a prediction threshold, denoted as in Equation (2),

$$AUC(f(\theta)) = \int_0^1 TPR(FPR(f_{\iota}(\theta)))d\iota \quad (2)$$

where ι represents the prediction threshold applied to model predictions, and TPR and FPR represent the true positive rate and false positive rates, respectively. AUC scores are limited to a range of zero to one, with a random predictor achieving an AUC of 0.5. Given the unbalanced distribution of the data, we used the AUC-ROC to assess the overall predictive accuracy of our models in accurately capturing dropout patterns.

2. ΔAUC : The increment in AUC-ROC is defined to measure changes in predictive performance as shown in Equation (3):

$$\Delta AUC(T, T') = AUC(T) - AUC(T') \quad (3)$$

where $AUC(T')$ refers to the AUC of an individual model, and $AUC(T)$ refers to a model trained with another approach. This allowed us to compare, for example, how AUC values were affected by different approaches. If ΔAUC was close to zero, we could conclude that the model performs approximately the same as individual models, but if it was positive or negative, the model performed better or worse, respectively, in context T relative to T' .

Next, we show the results of executing 10 cross-validations of the individual, centralized, transfer (direct, repurposed, voting, and stacked) and federated learning approaches. We evaluated the performance of each approach separately for each target entity's dataset. This also allowed us to demonstrate how model performance varied based on the size of the target datasets.

Figures 13–15 show the results for each destination entity in our study, demonstrating the AUC (area under the curve) and ΔAUC (increment of AUC) metrics. Our visual representation includes bar charts illustrating the average AUC achieved by each approach. Box plots are also used to display the distribution of ΔAUC values, centered around zero, allowing a clearer visualization of improvements or declines.

The left side of Figure 13 shows the results of the different approaches in terms of AUC when the first dataset (in our case the biggest) was the target. In that case, the baseline was its own individual/local model trained on its own data giving a mean value for the AUC of 0.8166 ± 0.0042 . The AUC values ranged between 0.819 and 0.746 which were the lowest of the three target datasets. The centralized model produced a similar but lower performance than the baseline, with an AUC of 0.8109 ± 0.0042 . Direct transfer learning models performed below the baseline, 0.7896 ± 0.0049 with the model transferred from the second dataset, and even lower with 0.7461 ± 0.0027 from the third dataset. This is because these models may be biased by the source data and had fewer samples to learn from, losing generalization on the first dataset which was the largest one. However, when these models were retrained on the target data, the results of the repurposed transfer improved, and they were better than using the individual model, with values of 0.8175 ± 0.0033 and

0.8168 ± 0.0040 from the second and third entities, respectively. The stacking approach, which combined the results of the individual models, performed very well and had a higher AUC than the baseline at 0.8196 ± 0.0017 . However, the soft voting and federated learning approaches produced lower results, performing below the baseline.

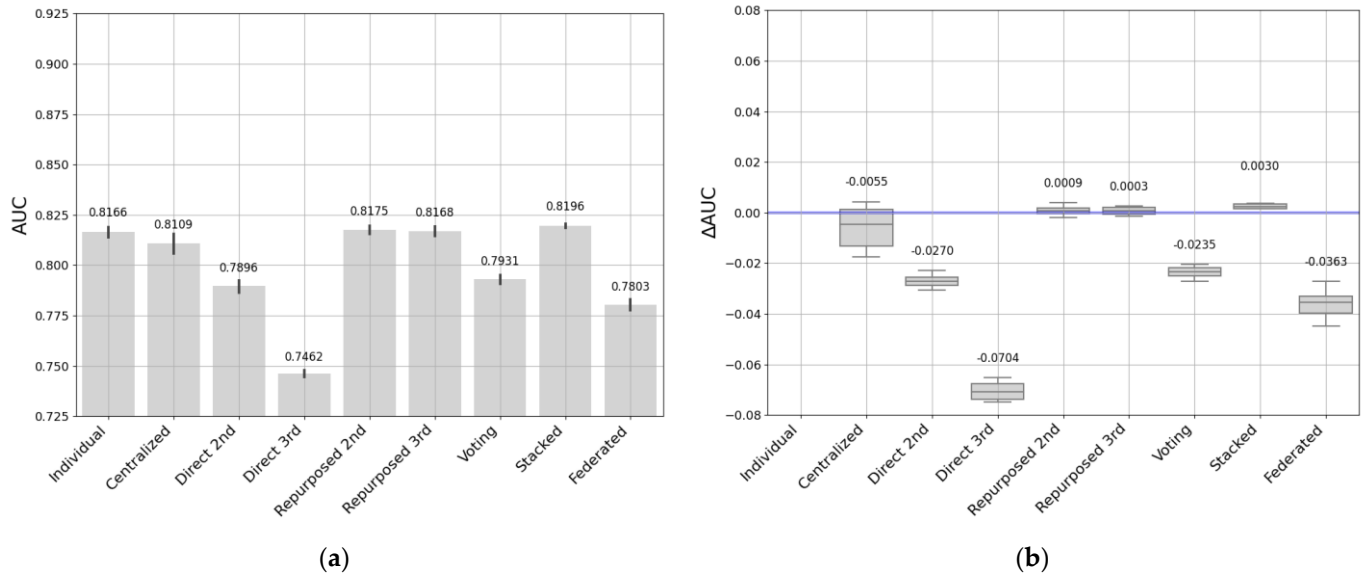


Figure 13. (a) AUC and (b) ΔAUC for all approaches. Target: 1st entity dataset.

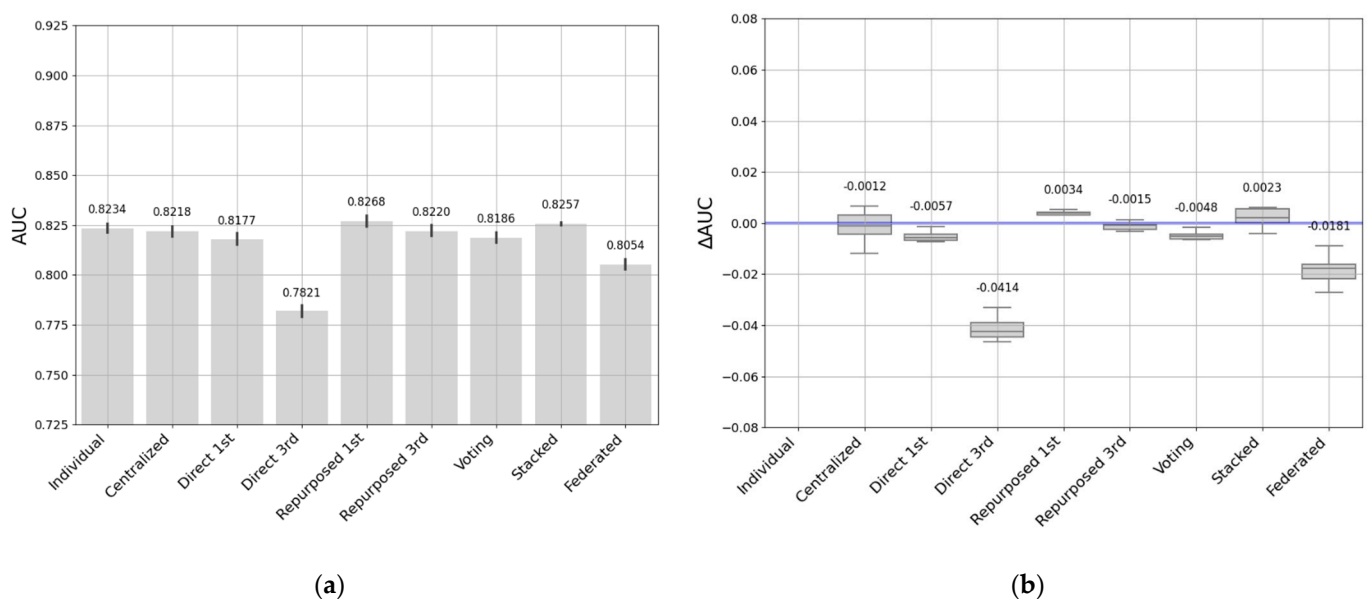


Figure 14. (a) AUC and (b) ΔAUC for all approaches. Target: 2nd entity dataset.

The right side of Figure 13 shows the increase or decrease in AUC for the different approaches when the target was the first dataset. All the boxes above the zero line indicate a model performing better than the baseline, which was the individual model for the first dataset (the largest). The results confirmed the previous AUC results; the performance of the repurposed models and the stacking approach were better than the baseline with very small differences in terms of ΔAUC . More specifically, the stacking approach stood out for its capability and potential since it offered better performance than all the other approaches with an increase of 0.3% in AUC. The repurposed models also performed very well compared with the single model, with increases of 0.09% and 0.03% when data came from the second individual model and the third individual model, respectively. The

centralized model produced a small decrease of -0.55% . The lowest results were produced by a direct transfer from individual model 3 (the smallest) with a decrease of -7.04% , along with the voting approach and the federated approach, with decreases of 2.35% and 3.63% , respectively.

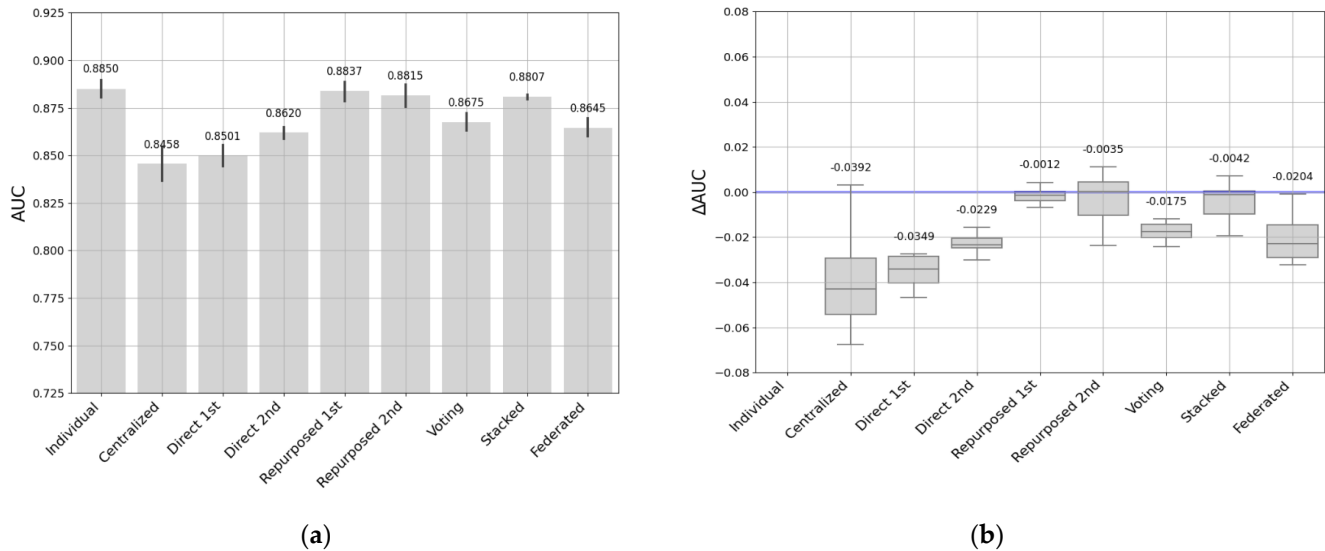


Figure 15. (a) AUC and (b) ΔAUC for all approaches. Target: 3rd entity dataset.

With the second (medium-sized) dataset as the target, as shown on the left side of Figure 14, the results of AUC were similar to the results for the first dataset (AUC ranging between 0.8234 and 0.7821), albeit with some differences. The main difference was that the baseline (individual model trained with the second dataset) did not produce the best results, with a value of 0.8234 ± 0.0037 . The centralized model performed similarly to the reference/baseline model, confirming its potential. The model transferred directly from the first, large dataset worked well (0.8177 ± 0.0044) but the model transferred from the third, small dataset worked worse (0.7821). The reused models performed very well, at the cost of retraining. In fact, the best results came from the repurposed first dataset. The soft-voting approach improved a little on the previous dataset with 0.8186 ± 0.0040 and the stacking method continued to improve with 0.8257 ± 0.0011 . Finally, the federated learning approaches continued to underperform, but improved a little by now taking advantage of learning from the largest dataset, with a value of 0.8054 ± 0.0039 . Looking at the increases or decreases in AUC for the second (medium-sized) dataset, the right side of Figure 14 shows that the results were similar to the results from the first dataset, with repurposed (0.34% from the first) and stacking (0.23%) approaches giving the best results, even outperforming the individual models. The centralized model produced a very small drop of -0.12% in AUC (lower than with the first dataset). The lowest results were from direct transfer from individual model 3 (the smallest), with a drop of -4.14% , along with the voting approach and the federated approach with drops of -0.48% and -3.63% , respectively.

The left side of Figure 15 shows the AUC results when the target was the third dataset, which was the smallest, with only 16,000 examples. The results were different and higher than those of the two previous target datasets. The AUC values ranged between 0.8850 and 0.8458. These results showed that in that dataset, the local bias was higher. This is reflected in the performance of the baseline, which was the best model, with an AUC of 0.8849 ± 0.0075 . The repurposed and stacked transferred learning models again produced the highest AUC values of about 0.88. However, the centralized approach produced the lowest AUC values (0.84), showing that it did not work very well when applied to a small target dataset. The model transferred directly from the first, large dataset produced the second lowest value (0.85), and the model transferred from the second, medium dataset

worked a little better (0.86). Finally, the voting and federated approaches produced very similar results with AUC values of about 0.86.

The right side of Figure 15 shows the increase or decrease in AUC for the third target dataset. In that case, all the approaches showed a decrease in AUC. The largest drop was produced by the centralized (−3.92%) and direct approaches (−3.49% and −2.29%). In contrast, the repurposed approach performed well on that dataset with a drop of only −0.12% from the first dataset and −0.35% from the second. The stacked approach also gave good results with a drop of −0.42%. Furthermore, the soft-voting and federated-voting models gave even worse results but with a smaller difference (−1.75% and −2.04%, respectively) than for the two previous cases using entities 1 and 2 as targets).

Finally, we carried out two tests to verify whether the differences in AUC were statistically significant. To achieve that, we conducted a paired t-test and a z-test with $\alpha = 0.05$ (see Table 2) to analyze significant differences between the performance of the approaches evaluated versus our baseline (individual model) in terms of AUC, using the individual course dataset as a target. A p -value of 0.05 or less indicated that the difference was significant. We used these two specific tests because they had been proposed as statistical analysis in other related works [15,16] to find any significant differences between these machine learning approaches.

Table 2. Test results.

Target	Source	Approach	Z-Test (p -Value)	T-Test (p -Value)
1st	1st + 2nd + 3rd	Centralized	0.073	0.090
	2nd	Direct transfer	<0.001 *	0.015 *
	3rd	Direct transfer	<0.001 *	<0.001 *
	2nd	Repurposed transfer	0.109	0.164
	3rd	Repurposed transfer	0.710	0.732
	2nd + 3rd	Voting	<0.001 *	<0.001 *
	1st + 2nd + 3rd	Stacking	0.021 *	0.046 *
	1st + 2nd + 3rd	Federated	<0.001 *	<0.001 *
2nd	1st + 2nd + 3rd	Centralized	0.623	0.569
	1st	Direct transfer	<0.001 *	<0.001 *
	3rd	Direct transfer	<0.001 *	<0.001 *
	1st	Repurposed transfer	0.106	0.120
	3rd	Repurposed transfer	0.623	0.087
	1st + 3rd	Voting	<0.001 *	<0.001 *
	1st + 2nd + 3rd	Stacking	0.035 *	0.065 *
	1st + 2nd + 3rd	Federated	<0.001 *	<0.001 *
3rd	1st + 2nd + 3rd	Centralized	<0.001 *	<0.001 *
	1st	Direct transfer	<0.001 *	<0.001 *
	2nd	Direct transfer	<0.001 *	<0.001 *
	1st	Repurposed transfer	0.253	0.307
	2nd	Repurposed transfer	0.317	0.368
	1st + 2nd	Voting	<0.001 *	<0.001 *
	1st + 2nd + 3rd	Stacking	0.101	0.136
	1st + 2nd + 3rd	Federated	<0.001 *	<0.001 *

* Significant differences.

As Table 2 shows, the p -values of the two tests (Z-test and T-test) confirmed the same results. The only approaches that produced significant differences in the three target dataset cases were direct transfer, soft voting, and federated learning. These approaches produced the lowest performance in AUC, and these differences with the baseline model were statistically significant indicating a real drop in performance.

With the other approaches, we identified no significant differences when using the centralized approach compared to the individual model in the first and second entities ($p > 0.05$), demonstrating that merging data from numerous entities does not decrease

performance in most cases, regardless of differences in data size. But when the target dataset was the smallest (third dataset), there were significant differences.

Looking at the stacking strategy, Table 2 shows that there were significant differences when comparing with the individual model with the first and second datasets as targets. In that case, stacking outperformed the baseline. But when the target dataset was the smallest (third dataset), there were no significant differences. In this case, the stacking approach produced a very similar performance with no significant differences.

Finally, repurposed transfer did not produce substantial differences in any of the cases, indicating that its performance was almost indistinguishable from the local models, even when the entities worked with different size datasets. Thus, it produced a very good performance in AUC on the three datasets or entities without any significant differences from the baseline models.

5. Discussion

Our experiments in this case study indicate that the different machine learning approaches produced subtle differences in performance with respect to the baseline local models. Additionally, each method had its own set of advantages and disadvantages. Therefore, entities or institutions must decide carefully in order to achieve a certain degree of improvement, or if they are interested in reducing effort and resources.

Overall, our findings underscore the effectiveness of the repurposed transfer and ensemble stacking approaches in terms of performance. These approaches showed high promise and consistently outperformed our baseline. The stacking technique especially stood out, significantly improving on the performance of individual models in almost all cases. However, we note that this technique did not clearly improve performance when working with a small dataset as the target, suggesting that its application may depend on the size and/or context of the available data. The repurposed model also provided notable results in all tests, outperforming the benchmark model in most cases. However, its performance may decrease on smaller datasets, which could be due to a lack of sufficient data for the pretrained model to adequately fit local features. These two approaches, stacking and repurposed, improved the results of the individual models, but it is important to bear in mind that these methods include extra resources, as they require the implementation of additional models or extra training for the transfer process.

On the other hand, while the centralized approach generally produced good results, it did not always outperform the individual model, for instance when applied to a smaller dataset as the target. This may be because the diversity and complexity of the data in larger datasets may exceed the generalization capabilities of the centralized model. It exhibited improved performance when used on the medium-size dataset but faced challenges when applied to smaller-size datasets, where differences between entities may have influenced its ability to generalize.

The least positive results were from the direct transfer approach, which was expected, as the entities were different. However, it is interesting to note that direct transfer performed remarkably well when transferring data from the largest dataset to the second dataset, suggesting that direct transfer can be effective in certain cases.

As for the voting approach, while it did not perform universally well, it did perform relatively well when applied to medium- or small-size datasets as the targets. This may be because the differences between individual models were less noticeable, and the weighted aggregation of models may have improved the robustness of the final result.

Although federated learning demonstrated a low level of performance, it is a viable option, especially in contexts where the privacy of student data must be preserved. Its ability to outperform direct transfer in terms of performance reinforces its importance, although its performance was found to be more stable on small- and medium-sized datasets.

Our results confirm the previously obtained results in different related works that have compared different machine learning approaches. In some studies [18,21,27], the performance of direct transfer learning models against individual models has been exploited.

These studies reveal that direct transfer methods retain most of their predictive power when contexts are similar, but they do not match the performance of individual models. Some works [17,20] compared individual, direct, and centralized approaches, showing that a centralized approach obtained better results than the direct transfer, but individual models were the best ones. Individual, direct transfer, soft voting, and stacking ensemble transfer were examined in [16], where similar results to individual models were obtained with soft voting and stacking, with a particular mention for the stacking method, which often exhibited improvements. In [15], individual, direct transfer, and repurposed transfer were compared. The results showed that repurposed transfer learning that retrained a previously created model with another dataset provided the best results and a significant accuracy improvement. Finally, other studies compare the innovative federated learning approach versus the individual and centralized, as seen in [14,25], concluding that individual models tend to achieve high accuracy, while centralized learning may have more dispersion, and federated learning even more.

Our research consolidates and extends the discussions present in the literature. However, our review is the only one that compares together all these approaches (individual learning, direct transfer learning, repurposed transfer learning, soft-voting and stacking ensemble transfer learning, and federated learning) using different sizes of datasets, providing a specific guide according to data availability. In summary, direct transfer and soft-voting zero-shot approaches are suggested when an institution do not have data. When an institution begins to obtain data and has small-size datasets, they can use pretrained models, such as repurposed transfer learning, obtaining very good performance. If an institution has larger quantities of data, maintaining and training an individual model is the primary option, as well as considering the stacking approach to improve collaborations between institutions and predictive capability.

Finally, we provide some general key findings derived from our experimentation:

- When there are privacy concerns, the most immediately appealing alternative to individual models seems to be federated learning. However, this approach never improves on the baseline. There are some transfer approaches (such as stacking or repurposed transfer learning) that can be used in those cases since they improve on the baseline and do not have to share data.
- In terms of performance, it seems better to train models on the source context and continue with the training process on the target (repurposed) than to merge all data together in a centralized approach.
- When the resources needed for each approach are important, one positive aspect of centralized and federated learning is that only one model is trained and maintained. However, other approaches—not based on external centralized servers, such as stacking or repurposed transfer learning—need to deal with a large number of models, since the source models need to be created and more importantly, infrastructure and resources are needed for training and managing the models in the target contexts, which may lead to high resource consumption there.

6. Conclusions and Future Work

The prediction of student dropout is considered one of the main tasks in the field of educational data mining (EDM) and learning analytics (LA). This study investigated how different machine learning approaches could address this problem when the data came from three different and different-size online educational entities by using homogeneous student activity data. Our main conclusion following the study is that in general, there are few differences in performance (AUC and Δ AUC), and some machine learning approaches perform better than others in all cases even though the target entity changes in size. Below, we summarize these differences, together with the advantages and disadvantages of each approach and some specific key findings from our experiments:

- The individual models fit the data well at each entity. However, this requires training as many individual models as there are entities. Collecting data takes time and

resources to maintain so many models. We found that these models tended to have better performance on smaller datasets, and performance may drop when entities have larger datasets.

- Centralized models can build robust, generalizable models by combining data from multiple educational entities. However, in the process of data aggregation, it is critical to address privacy concerns and biases. In addition, the size of the data can influence the effectiveness of this approach: when applied to a larger dataset as a target, it often yields better results, while when applied to smaller or more specific datasets it can lead to decreased performance. It is a good option when data from different entities can be combined, so that these entities do not have to work to create and maintain predictive models.
- Direct transfer has been shown to be feasible and useful in situations where the previously trained model is accurate and adequately matches the particularities of the target data. Direct transfer approaches may be influenced by the sizes of the source and target datasets, and in cases where the datasets are more similar, it seems to produce good results and minimize the duplication of effort between entities.
- Repurposed transfer learning produced the best results for different sizes of entity datasets. This approach improves performance by being specialized on the target dataset without the need to have a large quantity of data. Nevertheless, the entity must have the resources to maintain this model.
- Soft-voting models achieved good performance on medium- and small-sized entities, while there were greater performance differences with the largest dataset. It would be a good option when there are similarities in data sizes and context, and when entities are looking to improve performance by sharing models rather than sharing all their data.
- The stacked transfer technique produced robust, consistent results in all our experiments. It outperformed the individual models, demonstrating its effectiveness in improving model generalizability and adaptability, allowing us to learn and benefit from the strengths of each individual model. This approach improves performance when entities can collaborate effectively by combining predictions from their individual models.
- Federated learning addresses privacy and security issues without sharing critical data. Although there may be a slight decrease in performance, this approach stands out for its resilience and potential in environments where privacy is a priority. Similar to other approaches, the performance of federated learning can also be influenced by data size. It typically performs well with small- to medium-sized datasets while maintaining data privacy but may face challenges with large datasets due to communication and resource constraints.

The results obtained in our research can be generalized to other educational contexts. Our study is based on course data from three institutions managed by the same online course provider, so they are expected to have a high similarity among them. In this regard, our data were based on features about the interaction of the students with the LMS (learning management systems), which makes it easier for any institution to obtain similar features from any online educational environment such as Moodle, Canvas, Google Classroom or MOOCs (massive open online courses). This is a positive aspect in terms of generalizability. That is, the results obtained in our research can be extrapolated to educational contexts similar to the one presented in our paper, in terms of data used and educational environment.

With respect to the practical implications of the findings for educational institutions, our study constitutes a useful preliminary guide for selecting the most appropriate machine learning approaches from the point of view of a target institution and taking into account the size of its dataset:

- If the institution does not have their own data (labeled) to train and maintain a model, in this case, they should use zero-shot techniques such as direct transfer or voting,

although both obtain worse prediction results than an individual model. In the case where a partner institution has a similar context, we would choose direct transfer or combine the capabilities of different models using the soft-voting approach.

- If the institution has collected a few data (small-size dataset), individual models would not be the best option because they may not optimally converge. The best option is to use pretrained models from other institutions as a starting point and adapt them to the institution's local data. Among the most promising options are repurposed transfer and stacking transfer. We also mention the federated learning approaches which, although they obtained worse results, show a promising result to be studied in the future. We also note that the use of a centralized model can generate a greater dispersion and be biased by data from larger institutions.
- If the institution has a great quantity of data (medium- and large-size dataset), the main recommendation is to maintain and train an individual model, since approaches such as centralized and repurposed approaches do not offer significant advantages in terms of performance. However, as a strategic alternative, a stacked ensemble is highlighted as demonstrating superior performance and can be considered especially useful when institutions look for collaborations with others, even when the individual model is already efficient.

Finally, several future lines of research are envisaged that can deepen and expand our case study for predicting student dropout using multiple datasets or entities:

- Dropout timestamp: In our research, we did not consider the specific timestamp at which the dropout occurred. In particular, in our datasets, the students could start and finish courses at any time, so it was hard to identify the exact dropout time, as it was not a feature provided to us. In fact, the online course provider manually labeled dropouts after dropout, once the student was contacted to confirm dropping out. As a future line of research, it would be interesting to identify that moment (preferably by the use of any automatic method), as long as the dataset can provide it. The use of that new indicator could be of great power to analyze aspects such as the connection of dropouts with the difficulty weight of the modules in the courses, to identify dropouts as early as possible, to identify the topics inside a course that have more influence on dropouts, or to detect similar dropout behaviors among students. All of them would represent pieces of knowledge of great value that could be used to design interventions to effectively prevent dropout.
- Model scalability and efficiency: As data volumes continue to rise, model scalability and efficiency become increasingly important. Future research efforts could involve extensive experiments with more entities of different sizes and domains, for example, different education levels, not only higher education. This may help produce deeper insights into how dataset size and the specific domain impact the performance of predicting student dropout models. In our research, we employed data based on the activity levels of the student in the LMS Moodle system. As a future line of research, it would be interesting to study the performance of our approaches in other LMS systems, since some studies have shown that student dropout and performance may also depend on the LMS system used [30], and it could be a new variable to increase the knowledge of our models.
- Exploration of heterogeneous and multimodal data: Although this study focused on homogeneous data, future research could consider including heterogeneous data, where entities or datasets have different rather than identical attributes. An exploration of heterogeneous data could provide a broader, more accurate view of student behavior and characteristics. In this study, we used only attributes related to students' activities. The inclusion of multimodal attributes, encompassing text, video, and audio, obtained from sources such as forum postings and recordings, has the potential to enrich the predictive modeling process by capturing a broader range of student interactions and experiences.

Author Contributions: Conceptualization, S.V. and C.R.; data curation, J.M.P.; formal analysis, J.A.L.; funding acquisition, S.V.; methodology, S.V. and C.R.; project administration, S.V. and C.R.; software, J.M.P.; supervision, S.V.; validation, S.V. and C.R. visualization, J.M.P. and J.A.L.; writing—original draft, J.M.P., J.A.L., S.V. and C.R.; writing—review and editing, J.M.P., J.A.L., S.V. and C.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Project CTA-22/1085 entitled Learning Analytics for Improving the Quality of Teaching in Educational Settings, by the Project PID2020-115832GB-I00 project of Spanish Ministry of Science and Innovation and the European Regional Development Fund, and the ProyExcel-0069 project of the Andalusian University.

Data Availability Statement: The data are not publicly available due to the regulations of the private company who provided the online courses.

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

References

1. Keshavamurthy, U.; Guruprasad, H.S. Learning analytics: A survey. *Int. J. Comput. Trends Technol.* **2015**, *18*, 6. [\[CrossRef\]](#)
2. Romero, C.; Ventura, S. Educational data mining and learning analytics: An updated survey. *WIREs Data Min. Knowl. Discov.* **2020**, *10*, e1355. [\[CrossRef\]](#)
3. Smith, B.G. E-learning Technologies: A Comparative Study of Adult Learners Enrolled on Blended and Online Campuses Engaging in a Virtual Classroom. Ph.D. Thesis, Capella University, Minneapolis, MN, USA, 2010.
4. Dalipi, F.; Imran, A.S.; Kastrati, Z. MOOC dropout prediction using machine learning techniques: Review and research challenges. In Proceedings of the IEEE Global Engineering Education Conference (EDUCON'18), Santa Cruz de Tenerife, Spain, 17–20 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1007–1014. [\[CrossRef\]](#)
5. Prenkaj, B.; Velardi, P.; Stilo, G.; Distanto, D.; Faralli, S. A survey of machine learning approaches for student dropout prediction in online courses. *ACM Comput. Surv.* **2020**, *53*, 1–34. [\[CrossRef\]](#)
6. Solomon, D. Predicting performance and potential difficulties of university students using classification: Survey paper. *Int. J. Pure Appl. Math.* **2018**, *118*, 2703–2707.
7. Márquez-Vera, C.; Cano, A.; Romero, C.; Noaman, A.Y.M.; Fardoun, H.M.; Ventura, S. Early dropout prediction using data mining: A case study with high school students. *Expert Syst.* **2016**, *33*, 107–124. [\[CrossRef\]](#)
8. Donoso-Díaz, S.; Iturrieta, T.N.; Traverso, G.D. Sistemas de Alerta Temprana para estudiantes en riesgo de abandono de la Educación Superior. *Ens. Avaliação Políticas Públicas Em Educ.* **2018**, *26*, 944–967. [\[CrossRef\]](#)
9. Shafiq, D.A.; Marjani, M.; Habeeb, R.A.A.; Asirvatham, D. Student Retention Using Educational Data Mining and Predictive Analytics: A Systematic Literature Review. *IEEE Access* **2022**, *10*, 72480–72503. [\[CrossRef\]](#)
10. Xing, W.; Du, D. Dropout prediction in MOOCs: Using deep learning for personalized intervention. *J. Educ. Comput. Res.* **2019**, *57*, 547–570. [\[CrossRef\]](#)
11. Miao, Q.; Lin, H.; Hu, J.; Wang, X. An intelligent and privacy-enhanced data sharing strategy for blockchain-empowered Internet of Things. *Digit. Commun. Netw.* **2022**, *8*, 636–643. [\[CrossRef\]](#)
12. Gardner, J.; Yang, Y.; Baker, R.S.; Brooks, C. Modeling and Experimental Design for MOOC Dropout Prediction: A Replication Perspective. In Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019), Montréal, QC, Canada, 2–5 July 2019.
13. Fauzi, M.A.; Yang, B.; Blobel, B. Comparative Analysis between Individual, Centralized, and Federated Learning for Smartwatch Based Stress Detection. *J. Pers. Med.* **2022**, *12*, 1584. [\[CrossRef\]](#)
14. Fachola, C.; Tornaría, A.; Bermolen, P.; Capdehourat, G.; Etcheverry, L.; Fariello, M.I. Federated Learning for Data Analytics in Education. *Data* **2023**, *8*, 43. [\[CrossRef\]](#)
15. Tsiakmaki, M.; Kostopoulos, G.; Kotsiantis, S.; Ragos, O. Transfer Learning from Deep Neural Networks for Predicting Student Performance. *Appl. Sci.* **2020**, *10*, 2145. [\[CrossRef\]](#)
16. Gardner, J.; Yu, R.; Nguyen, Q.; Brooks, C.; Kizilcec, R. Cross-Institutional Transfer Learning for Educational Models: Implications for Model Performance, Fairness, and Equity. In Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, Chicago, IL, USA, 12–15 June 2023. [\[CrossRef\]](#)
17. Vitiello, M.; Walk, S.; Chang, V.; Hernandez, R.; Helic, D.; Guetl, C. *Mooc Dropouts: A Multi-System Classifier*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 300–314. [\[CrossRef\]](#)
18. Jayaprakash, S.M.; Moody, E.W.; Lauría, E.J.M.; Regan, J.R.; Baron, J.D. Early alert of academically at-risk students: An open source analytics initiative. *J. Learn. Anal.* **2014**, *1*, 6–47. [\[CrossRef\]](#)
19. Li, X.; Song, D.; Han, M.; Zhang, Y.; Kizilcec, R.F. On the limits of algorithmic prediction across the globe. *arXiv* **2021**, arXiv:2103.15212.
20. Ocumpaugh, J.; Baker, R.; Gowda, S.; Heffernan, N.; Heffernan, C. Population validity for educational data mining models: A case study in affect detection. *Br. J. Educ. Technol.* **2014**, *45*, 487–501. [\[CrossRef\]](#)

21. López-Zambrano, J.; Lara, J.A.; Romero, C. Towards portability of models for predicting students' final performance in university courses starting from moodle Logs. *Appl. Sci.* **2020**, *10*, 354. [[CrossRef](#)]
22. Smietanka, M.; Pithadia, H.; Treleaven, P. Federated learning for privacy-preserving data access. *Int. J. Data Sci. Big Data Anal.* **2021**, *1*, 1. [[CrossRef](#)]
23. McMahan, H.B.; Moore, E.; Ramage, D.; Agüera y Arcas, B. Federated Learning of Deep Networks Using Model Averaging. *arXiv* **2016**, arXiv:1602.05629.
24. Peng, S.; Yang, Y.; Mao, M.; Park, D. Centralized Machine Learning Versus Federated Averaging: A Comparison using the MNIST Dataset. *KSII Trans. Internet Inf. Syst.* **2022**, *16*, 742–756. [[CrossRef](#)]
25. Guo, S.; Zeng, D.; Dong, S. Pedagogical Data Analysis via Federated Learning toward Education 4.0. *Am. J. Educ. Inf. Technol.* **2020**, *4*, 56–65.
26. He, J.; Bailey, J.; Rubinstein, B.; Zhang, R. Identifying at-risk students in massive open online courses. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29. [[CrossRef](#)]
27. Whitehill, J.; Mohan, K.; Seaton, D.; Rosen, Y.; Tingley, D. Delving deeper into MOOC student dropout prediction. *arXiv* **2017**, arXiv:1702.06404.
28. Porras, J.M.; Porras, A.; Fernández, J.; Romero, C.; Ventura, S. *Selecting the Best Approach for Predicting Student Dropout in Full Online Private Higher Education*; LASI: Singapore, 2023.
29. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [[CrossRef](#)]
30. Kanetaki, Z.; Stergiou, C.; Bekas, G.; Troussas, C.; Sgouropoulou, C. The impact of different learning approaches based on MS Teams and Moodle on students' performance in an on-line mechanical CAD module. *Glob. J. Eng. Educ.* **2021**, *23*, 185–190.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.