CTU Journal of Innovation and Sustainable Development
ISSN 2615-9422 | e-ISSN 2815-5602

DOI:10.22144/ctujoisd.2023.039

Impact of dimensionality reduction techniques on student performance prediction using machine learning

Koushik Roy[1*], Huu-Hoa Nguyen[2], and Dewan Md. Farid[1]

[1,3]*United International University, Dhaka, Bangladesh*

[2]*College of Information and Communication Technology, Can Tho University, Viet Nam*

**Corresponding author (rkoushikroy2@gmail.com)*

Article info.

Received 19 Jul 2023
Revised 17 Sep 2023
Accepted 2 Oct 2023

Keywords

Backward elimination, dimensionality reduction, forward selection, recursive feature elimination, student performance prediction

ABSTRACT

This study addresses the crucial issue of predicting student performance in educational data mining (EDM) by proposing an Adaptive Dimensionality Reduction Algorithm (ADRA). ADRA efficiently reduces the dimensionality of student data, encompassing various academic, demographic, behavioral, social, and health-related features. It achieves this by iteratively selecting the most relevant features based on a combined normalized mean rank of five feature ranking methods. This reduction in dimensionality enhances the performance of predictive models and provides valuable insights into the key factors influencing student performance. The study evaluates ADRA using four different student performance datasets and six machine learning algorithms, comparing it to three existing dimensionality reduction methods. The results show that ADRA achieves an average dimensionality reduction factor of 6.2 while maintaining comprable accuracy with other mehtods.

1. INTRODUCTION

In today's era of rapid advancements in computer technology, communication systems, and sensors, a vast amount of data is generated by individuals or devices across different domains. This abundance of data has led to the emergence of the big data era (Romero & Ventura, 2020). In the field of education, various digital learning platforms, such as student information systems, learning management systems, massive open online courses, and virtual learning systems, generate substantial volumes of data. However, effectively analyzing this educational data and extracting valuable insights from it presents a significant challenge for researchers and educational institutions. The goal is to uncover hidden information that can benefit learners, instructors, and administrators alike (Zhang et al., 2023).

In the field of educational data mining (EDM), accurately predicting student performance is a crucial challenge. With the integration of technology in educational settings, an extensive range of student data is now available, encompassing not only academic factors but also demographic, behavioral, social, and health-related aspects. However, the abundance of input features can lead to computational complexities, overfitting, and reduced interpretability. To address these issues, there is a pressing need for efficient and automated dimensionality reduction techniques (Sabri et al., 2023; Xue & Niu, 2023). Therefore, this study introduces the Adaptive Dimensionality Reduction Algorithm (ADRA), which leverages a novel approach involving the combined normalized mean rank of five distinct feature ranking methods. Through an iterative process, ADRA selects the most relevant features to optimize cross-validation accuracy. By simplifying datasets and improving the fit of predictive models, this algorithm provides valuable insights to educators and policymakers, shedding light on the crucial factors influencing student performance.

This research aims to address two main research questions. The first research question is: How does the application of the ADRA impact the performance of machine learning algorithms in predicting student performance? By studying the impact of ADRA on performance, we can determine its effectiveness in enhancing the predictive capabilities of machine learning models. Furthermore, we aim to explore the key factors identified by ADRA that significantly influence students' academic performance and compare them to traditional feature selection techniques. This leads us to the second research question: What are the key factors identified by ADRA, and how do they compare to traditional feature selection techniques in terms of their impact on students' academic performance? Answering these questions will provide valuable insights into the effectiveness and uniqueness of ADRA in uncovering the most influential factors affecting student performance. Ultimately, this research aims to shed light on the impact of dimensionality reduction techniques on student performance prediction using machine learning algorithms (Hussain et al., 2022; Fida et al., 2022; Bilal et al., 2022; Yağcı, 2022).

In recent years, a significant body of research has focused on predicting student performance and identifying factors contributing to academic success using data mining and machine learning techniques. These studies employ diverse methodologies and highlight the potential of predictive analytics in understanding and improving student performance in educational settings (Ouyang et al., 2023; Yılmaz & Sekeroglu, 2019). One prominent area of research revolves around the application of various machine learning algorithms for student performance prediction (Shetu et al., 2021; VeeraManickam et al., 2019; Xue & Niu, 2023). Another line of research focuses on feature selection techniques to improve prediction models in the field of Educational Data Mining (EDM) (Estrera et al., 2017; Ramaswami & Bhaskaran, 2009; Febro, 2019; Zaffar et al., 2018; Alhassan et al., 2020; Mythili & Shanavas, 2014). Despite the existing literature on student performance prediction and feature selection, certain limitations persist. Many studies primarily focus on a specific algorithm or feature selection technique, leading to a fragmented understanding of the overall landscape. Furthermore, the scarcity of research that combines both dimensionality reduction and prediction modeling hampers the development of more comprehensive and accurate systems for student performance prediction. These gaps in the existing literature motivate the present study, which proposes an Adaptive Dimensionality Reduction Algorithm (ADRA) that combines the strengths of feature ranking methods and iteratively selects the best features.

2. DATASET DESCRIPTION

This study employs four educational datasets: the xAPI-Educational Mining Dataset (xAPI) retrieved from Kaggle in 2016 (Amrieh et al., 2016), the Secondary Student Performance Dataset (SSP) obtained from the UCI ML Archive in 2014 (Cortez, 2014), the Higher Education Student Performance Dataset (HESP) sourced from Kaggle (Sekeroglu, 2019), and the Student Performance Prediction - Western-OC2-Lab (WOC2) dataset acquired from GitHub in 2020 (Injadat et al., 2020). These datasets exhibit variability in terms of acquisition year, instance count, attribute count, and feature types. The xAPI dataset encompasses 480 instances with 16 attributes, including demographic, academic, and behavioral features, primarily suitable for classification tasks. In contrast, the SSP dataset comprises 1044 instances with 33 attributes, encompassing demographic, academic, behavioral, and health-related features, making it suitable for both classification and regression tasks. The HESP dataset contains 145 instances with 31 attributes, covering demographic, academic, and behavioral features, and it is applicable for classification and regression tasks. Lastly, the WOC2 dataset consists of 486 instances with 9 attributes, focusing mainly on academic features and classification tasks. These datasets represent valuable resources for exploring student performance and behavior in diverse educational contexts. Detailed comparison of these datasets is presented in Table 1.

3. METHOD

This research investigated the impact of dimensionality reduction techniques on student performance prediction using machine learning. The methodology involved data cleaning, exploratory data analysis, preprocessing, dimensionality reduction and evaluation. Multiple machine learning algorithms were employed and evaluated using cross-validation. The experimental results and findings are discussed in detail, with a visual representation of the methodology provided in Figure 1.

Table 1. Comparison among four different datasets used in this study

Dataset	Year	Source	Instance	Attributes	Feature	Associated Tasks
xAPI - Educational Mining Dataset (XAPI) (Amrieh et al., 2016)	2016	Kaggle	480	16	Demographic, Academic, Behavioral	Classification
Secondary Student Performance Dataset (SSP) (Cortez, 2014)	2014	UCI ML Archive	1044	33	Demographic, Academic, Behavioral, Health	Classification, Regression
Higher Education Student Performance Dataset (HESP) (Sekeroglu, 2019)	2019	Kaggle	145	31	Demographic, Academic, Behavioral	Classification, Regression
Student Performance Prediction - Western-OC2-Lab (WOC2) (Injadat et al., 2020)	2020	GitHub	486	9	Academic	Classification

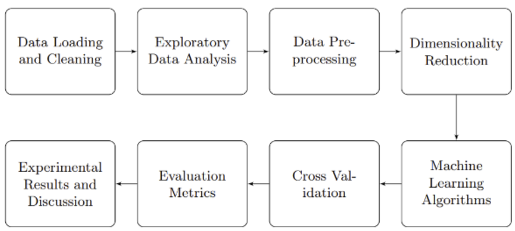

Figure 1. Block diagram of system method

3.1. Exploratory Data Analysis

In the initial phase of the methodology, the data was loaded and cleaned to ensure data integrity and reliability, which is crucial for machine learning projects. Missing values, duplicate entries, and columns with constant values were eliminated. This meticulous data cleaning process established a solid foundation for subsequent steps. An Exploratory Data Analysis (EDA) was conducted to gain insights into the dataset, including examining data distribution, understanding attribute types and meanings, visualizing data, identifying outliers, and assessing the relationship between variables and the target variable. The EDA process provided valuable insights, enabling informed decisions on data preprocessing, feature engineering, and model selection for improved effectiveness and accuracy in the study.

From the output class distribution of the four datasets, it is apparent that xAPI and HESP datasets are fairly balanced while SSP dataset is slightly imbalanced and WOC2 dataset is hevily imbalanced. This data distribution required us to use a data balancing method in the preprocessing stage.

3.2. Data preprocessing

In our study's preprocessing phase, we addressed several key issues to prepare the dataset for machine learning algorithms. One challenge was handling categorical data, which we transformed into numerical values using Label Encoders. This technique assigned unique categories with corresponding integers. The dataset was then split into training and testing sets with an 80-20 ratio, using stratified sampling to maintain representative distributions. To further optimize the data, we applied standardization, scaling the values around the mean with unit standard deviation. Additionally, we utilized ADASYN (Adaptive Synthetic) to address class imbalance. These preprocessing techniques ensured our data was ready for dimensionality reduction and application of machine learning algorithms.

To ensure consistent evaluation among datasets, the SSP and HESP datasets underwent a transformation using the five-number method. The class count was divided into three categories: "Low," "Mid," and "High." The minimum value to the first quartile was classified as "Low," the range between the first and third quartiles as "Mid," and the third quartile to the maximum value as "High." This transformation created three distinct classes based on class count ranges, facilitating consistent evaluation and analysis across the datasets.

3.3. Dimensionality Reduction Method

Dimensionality reduction plays a pivotal role in optimizing the performance of machine learning models by reducing the complexity of the input space. In this section, we present a collection of feature selection algorithms, namely Recursive

Feature Elimination (RFE), Forward Selection (FS), Backward Elimination (BE), and our novel method, the Adaptive Dimensionality Reduction Algorithm (ADRA). Initially, we describe the existing algorithms and subsequently introduce ADRA as an innovative approach. All of these approaches are based on Wrapper methods, which involve feature selection guided by a specific machine learning algorithm. This methodology employs a greedy search strategy, where various feature combinations are evaluated based on a specific criterion.

3.3.1. Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) is a feature selection algorithm that recursively eliminates the least informative features based on a specified criterion. The algorithm starts with the entire feature set, trains a model, ranks the features, and eliminates the lowest-ranking feature. This process is repeated until the optimal number of features is determined (Algorithm 1).

Algorithm 1. Recursive Feature Elimination (RFE)

```
Input: DataFrame X, Target Column y, Model,
       Number of Folds for Cross-Validation (cv)
Output: Selected Features

1. X ← Input DataFrame
2. y ← Target column of DataFrame
3. selected_features ← set of all features
4. n ← number of features in selected_features
5. optimal_score ← 0
6. optimal_feature_subset ← Ø

7. while true:
8.    worst_feature ← None
9.    for feature in selected_features:
10.        candidate_features ← selected_features -
           feature
11.       X_train_fs ← X[candidate_features]
12.       cv_accuracy_fs ← cross_val_score(Model,
          X_train_fs, y, cv, 'accuracy')
13.       avg_accuracy ← mean(cv_accuracy_fs)
14.       if avg_accuracy > optimal_score:
15.          optimal_score ← avg_accuracy
16.          optimal_feature_subset ←
             candidate_features
17.          worst_feature ← feature
18.   if worst_feature is not None:
19.       selected_features ← selected_features -
          worst_feature
20.       n ← n - 1
21.       if n ≤ 1:
22.          break
23.   else:
24.       break
```

3.3.2. Forward Selection (FS)

Forward Selection (FS) is a feature selection algorithm that incrementally builds a feature subset by iteratively adding the most informative features based on a specified criterion. The algorithm starts with an empty set and gradually adds features until a termination criterion is met (Algorithm 2).

Algorithm 2. Forward Selection

```
Input: DataFrame X, Target Column y, Model,
       Number of Folds for Cross-Validation (cv)
Output: Selected Features

1. X ← Input DataFrame
2. y ← Target column of DataFrame
3. selected_features ← Ø
4. n ← number of features in X
5. optimal_score ← 0
6. optimal_feature_subset ← Ø

7. while true:
8.    best_feature ← None

9.    for feature in X:
10.       if feature not in selected_features:
11.          candidate_features ← selected_features
             + feature
12.          X_train_fs ← X[candidate_features]
13.          cv_accuracy_fs ←
             cross_val_score(Model, X_train_fs, y, cv,
             'accuracy')
14.          avg_accuracy ← mean(cv_accuracy_fs)

15.          if avg_accuracy > optimal_score:
16.             optimal_score ← avg_accuracy
17.             best_feature ← feature

18.   if best_feature is not None:
19.       selected_features ← selected_features +
          best_feature
20.       n ← n - 1
21.       if n ≤ 0:
22.          break
23.   else:
24.       break
```

3.3.3. Backward Elimination (BE)

Backward Elimination (BE) is a feature selection algorithm that starts with the entire feature set and iteratively removes the least informative features based on a specified criterion. The algorithm gradually reduces the feature subset until a termination criterion is met (Algorithm 3).

Algorithm 3. Backward Elimination (BE)

Input: DataFrame X, Target Column y, Model, Number of Folds for Cross-Validation (cv)
Output: Selected Features

```
1. X ← Input DataFrame
2. y ← Target column of DataFrame
3. selected_features ← set of all features
4. n ← number of features in selected_features
5. optimal_score ← 0
6. optimal_feature_subset ← ∅

7. while true:
8.     best_feature ← None

9.     for feature in selected_features:
10.        candidate_features ← selected_features - feature
11.        X_train_fs ← X[candidate_features]
12.        cv_accuracy_fs ← cross_val_score(Model, X_train_fs, y, cv, 'accuracy')
13.        avg_accuracy ← mean(cv_accuracy_fs)

14.        if avg_accuracy > optimal_score:
15.            optimal_score ← avg_accuracy
16.            best_feature ← feature

17.    if best_feature is not None:
18.        selected_features ← selected_features - best_feature
19.        n ← n - 1
20.        if n ≤ 1:
21.            break
22.    else:
23.        break
```

3.3.4. *Adaptive Dimensionality Reduction Algorithm (ADRA)*

Our proposed Adaptive Dimensionality Reduction Algorithm (ADRA) is presented in Algorithm 4. ADRA combines multiple feature ranking methods, including Information Gain, Chi-Square Test, Mutual Information, Relief, and Gini Importance, to evaluate the relevance of each feature. The algorithm calculates feature scores using these methods, normalizes the ranks, and computes an average rank for each feature.

In the feature selection stage, ADRA iteratively builds the feature subset by selecting features based on their rank. Starting with the feature with the highest rank, ADRA sequentially adds one feature at a time and evaluates the cross-validation accuracy of the model. If the addition of a feature improves accuracy, it is retained; otherwise, it is removed from the selected features. This process continues until the best subset of features is identified, maximizing the model's performance.

Algorithm 4. Adaptive Dimensionality Reduction Algorithm (ADRA)

Input: Input DataFrame X, Target Column y
Output: Selected Features

```
1. X ← Input DataFrame excluding the Target column
2. y ← Target column of DataFrame
3. Calculate feature scores using Information Gain, Chi-Square Test, Mutual Information, Relief, and Gini Importance
4. Normalize each rank and calculate the average rank
5. Sort features based on the average rank in descending order
6. best_features ← sorted list of features
7. selected_features ← empty list
8. best_accuracy ← 0

9. for feature in best_features:
10.    selected_features.append(feature)
11.    X_train_fs ← X[selected_features]
12.    X_test_fs ← X_test[selected_features]
13.    cv_accuracy_fs ← cross_val_score(Model, X_train_fs, y_train, cv, 'accuracy')

14.    if mean(cv_accuracy_fs) > best_accuracy:
15.        best_accuracy ← mean(cv_accuracy_fs)
16.    else:
17.        selected_features.remove(feature)
```

The ADRA algorithm effectively reduces the dimensionality of the dataset by adaptively selecting the most informative features. By combining multiple feature ranking methods and an iterative selection strategy, ADRA ensures a comprehensive assessment of feature relevance, improving the accuracy and efficiency of machine learning models.

3.4. Comparison

Our proposed Adaptive Dimensionality Reduction Algorithm (ADRA) builds upon existing feature selection methods, including RFE, FS, and BE. However, ADRA introduces a novel approach that combines multiple feature ranking methods and an adaptive selection strategy to identify the most relevant features for a specific problem. This approach offers several advantages over the existing methods.

ADRA leverages the comprehensive evaluation of feature relevance provided by multiple ranking methods. By considering the average rank of each feature across these methods, ADRA ensures a more robust assessment of feature importance, leading to improved model performance.

Regarding dimensionality reduction, ADRA's adaptive selection strategy allows for the efficient identification of the best subset of features. By iteratively adding and removing features based on their impact on the model's accuracy, ADRA dynamically adapts to the specific problem, reducing the dimensionality of the dataset effectively.

Furthermore, ADRA considers the training time of the model as a crucial factor. By combining feature ranking methods and an iterative selection process, ADRA optimizes the feature selection procedure, reducing the computational complexity and speeding up the training process.

In summary, ADRA offers superior accuracy, efficient dimensionality reduction, and reduced training time compared to the existing feature selection methods. These advantages make ADRA a promising approach for enhancing the performance of machine learning models.

3.5. Cross Validation

Cross-validation is a statistical method used to evaluate learning algorithms by dividing the data into two parts: one for training the model and the other for validating it. This process ensures that each data point is included in the validation set at some point, increasing the reliability of the model evaluation. The most common form of cross-validation is k-fold cross-validation, where the data is divided into k subsets. Each iteration involves training the model on k-1 subsets and validating it on the remaining subset. This approach provides an unbiased estimate of model generalization and helps prevent overfitting. In this research, 10-fold cross-validation was used to thoroughly evaluate machine learning models for predicting student performance.

3.6. Machine Learning Algorithms

This study employed various machine learning algorithms to investigate the influence of dimensionality reduction techniques on predicting student performance. The algorithms used included Decision Tree Classifier (DT), Gaussian Naive Bayes (GNB), Multilayer Perceptron Classifier (MLP), Random Forest Classifier (RFC), Bagging Classifier (BC), and AdaBoost Classifier (ABC). categorical and continuous data. By utilizing a range of algorithms, the study aimed to identify the most effective model for student performance prediction. The findings of this study contribute to a better understanding of the impact of dimensionality reduction techniques on machine learning algorithm performance in the context of predicting student outcomes.

3.7. Evaluation

The model's performance was evaluated using metrics such as accuracy, F1-score, precision, recall and Dimensionality Reduction Factor (DRF), shown in Equation 1. These metrics provided a comprehensive evaluation of the model's ability to predict student performance in various scenarios. A 10-fold cross-validation approach was adopted to ensure reliable performance assessment. Overall, the evaluation of the machine learning models yielded valuable insights for improving educational outcomes in the field of educational data mining.

$$\text{DRF} = \frac{\text{Total dimension}}{Reduced\ dimension} \quad (1)$$

4. RESULTS AND DISCUSSION

In this section, we present the results of our experiments and provide a detailed discussion. We evaluate the performance of various methods and models using different datasets. The comparison among the datasets is shown in Table 2, and the performance of different models is presented in Table 3. Additionally, Figure 2 illustrates the accuracy of different methods, while Figure 3 depicts the performance in terms of the Dimensionality Reduction Factor (DRF). The accuracy and DRF presented in Figures 2, 3 and Table 2 are averaged over all six models. From Figure 2, it can be shown that the average accuracy is similar for all different methods studied in this research. Whereas 3 shows that our proposed ADRA method provides the best average DRF across all four datasets, highest among them is 7.3.

Table 2. Comparison among four different datasets used in this study

Dataset	Method	Accuracy	F1 score	Precision	Recall	DRF
XAPI	ADRA	0.75	0.75	0.75	0.75	2.05
XAPI	All Features	0.75	0.75	0.76	0.76	1
XAPI	RFE	0.75	0.76	0.77	0.77	1.1
XAPI	FS	0.75	0.76	0.77	0.77	1.48
XAPI	BE	0.75	0.77	0.77	0.77	1.45
SSP	ADRA	0.68	0.67	0.7	0.68	1.9
SSP	All Features	0.69	0.69	0.73	0.69	1
SSP	RFE	0.69	0.68	0.72	0.69	1.1
SSP	FS	0.69	0.69	0.71	0.69	1.42
SSP	BE	0.69	0.7	0.72	0.7	1.37
HESP	ADRA	0.58	0.58	0.59	0.58	7.35
HESP	All Features	0.45	0.44	0.46	0.44	1
HESP	RFE	0.45	0.49	0.51	0.5	1.3
HESP	FS	0.45	0.53	0.54	0.54	3.07
HESP	BE	0.45	0.54	0.55	0.54	2.15
WOC2	ADRA	0.92	0.92	0.93	0.92	1.6
WOC2	All Features	0.91	0.9	0.92	0.91	1
WOC2	RFE	0.91	0.91	0.92	0.91	1.2
WOC2	FS	0.91	0.93	0.93	0.93	1.2
WOC2	BE	0.91	0.93	0.93	0.93	1.28

To address the first research question, we evaluated the effectiveness of the ADRA method in enhancing the predictive capabilities of machine learning models. We compared the performance of various methods and models using different datasets. Table 2 provides a comprehensive comparison among the four datasets used in our study, highlighting the accuracy, f1_score, precision, recall, and Dimensionality Reduction Factor (DRF) for each combination of dataset and method.

Our findings indicate that the XAPI and SSP datasets consistently yield similar performance across all methods, with accuracy values ranging from 0.68 to 0.75. However, the HESP dataset shows lower accuracy values, ranging from 0.45 to 0.58. In contrast, the WOC2 dataset consistently achieves high average accuracy scores, around 0.91 for all methods.

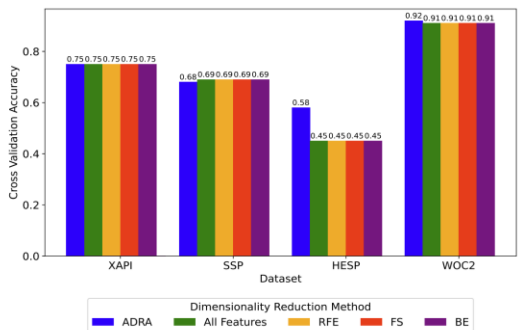

Figure 2. Accuracy comparison among different methods and datasets

Table 3. Comparison among six different models used in this study

Model	Accuracy	F1 score	Precision	Recall	DRF
DT	0.73	0.74	0.74	0.74	1.62
GNB	0.65	0.66	0.69	0.67	1.99
MLP	0.69	0.72	0.72	0.72	2.12
RFC	0.78	0.78	0.79	0.79	1.5
BC	0.76	0.77	0.78	0.77	1.61
ABC	0.64	0.65	0.68	0.66	1.68

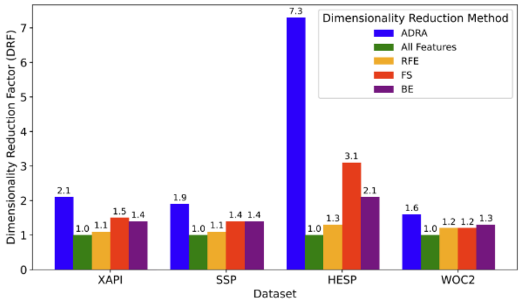

Figure 3. Comparison of Dimensionality Reduction Factor (DRF) among different methods

When considering the DRF, we observe that the ADRA method consistently outperforms other methods for all datasets. This indicates that ADRA effectively reduces the dimensionality of the datasets, allowing for more efficient and accurate predictions of student performance. Figure 3 visually demonstrates the superior DRF achieved by our proposed ADRA method across all four datasets.

The second research question aimed to compare the key factors identified by the ADRA method with traditional feature selection techniques. Six different models were analyzed, and their performance was assessed in predicting students' academic performance. The results, as shown in Table 3, revealed that the Random Forest Classifier (RFC) model achieved the highest accuracy score of 0.78, closely followed by the Boosted Classifier (BC) model with an accuracy score of 0.76. The Decision Tree (DT) and Multi-Layer Perceptron (MLP) models also performed well, with accuracy scores of 0.73 and 0.69, respectively. However, the Gaussian Naive Bayes (GNB) and AdaBoost Classifier (ABC) models exhibited lower accuracy scores of 0.65 and 0.64, respectively.

Regarding dimensionality reduction, the MLP model showed the highest average Dimensionality Reduction Factor (DRF) of 2.12, indicating effective reduction of input features while maintaining comparable evaluation metrics to other models. Overall, the findings highlight the effectiveness of the ADRA method in achieving higher DRF values and emphasize the RFC model as the top performer in terms of accuracy.

In conclusion, the research successfully addressed the second research question by comparing the ADRA method with traditional feature selection techniques. The results demonstrate the superiority of ADRA in achieving higher DRF values and highlight the RFC model as the most accurate predictor. These findings contribute to the field of educational data mining and provide insights for improving student performance prediction models.

5. CONCLUSION AND FUTURE WORK

In this research, we proposed an Adaptive Dimensionality Reduction Algorithm (ADRA) for student performance prediction using machine learning. Our algorithm combines multiple feature ranking methods and an iterative selection strategy to identify the most relevant features. We compared ADRA with existing methods and evaluated its performance on four student performance datasets.

The experimental results demonstrated that ADRA consistently outperformed other methods, achieving an average Dimensionality Reduction Factor (DRF) of 6.2 across all datasets. By effectively reducing the dimensionality while maintaining competitive accuracy, ADRA offers a practical and automated approach for feature selection in educational data mining. This can aid in optimizing learning paths and interventions to improve student outcomes.

The findings of this research have significant implications for the field. Future work can focus on refining and optimizing the ADRA method, incorporating additional feature ranking methods, and fine-tuning the selection strategy. Investigating the impact of dimensionality reduction techniques on different machine learning algorithms would provide valuable guidance for selecting suitable algorithms for student performance prediction tasks.

The generalizability of the ADRA method should be examined by conducting experiments with larger and more diverse datasets. Exploring its application in other educational domains such as student engagement, dropout prediction, and personalized learning would be valuable.

Moreover, further research can delve into the interpretability of the selected features, understanding their relationships with student outcomes. This can facilitate the development of targeted interventions and support systems.

In conclusion, our research presents the ADRA algorithm as an efficient approach for feature selection in student performance prediction. The findings and future directions outlined contribute to the ongoing efforts in leveraging machine learning to enhance educational outcomes.

REFERENCES

Amrieh, E. A., Hamtini, T., & Aljarah, I. (2016). Mining educational data to predict student's academic performance using ensemble methods. *International Journal of Database Theory and Application*, *9*(8), 119-136.

Alhassan, A., Zafar, B., & Mueen, A. (2020). Predict students' academic performance based on their assessment grades and online activity data. *International Journal of Advanced Computer Science and Applications*, *11*(4).

Bilal, M., Omar, M., Anwar, W., Bokhari, R. H., & Choi, G. S. (2022). The role of demographic and academic features in a student performance prediction. *Scientific Reports*, *12*(1), 12508.

Cortez, P. (2014). Student Performance. [Dataset]. *UCI Machine Learning Repository*. https://doi.org/10.24432/C5TG7T.

Estrera, P. J. M., Natan, P. E., Rivera, B. G. T., & Colarte, F. B. (2017). Student Performance Analysis for Academic Ranking Using Decision Tree Approach in University of Science and Technology of Southern Philippines Senior High School Abstract. *International Journal of Engineering and Technology*, *3*(5), 147-153.

Febro, J. D. (2019). Utilizing feature selection in identifying predicting factors of student retention. *International Journal of Advanced Computer Science and Applications*, *10*(9).

Fida, S., Masood, N., Tariq, N., & Qayyum, F. (2022). A Novel Hybrid Ensemble Clustering Technique for Student Performance Prediction. *JUCS: Journal of Universal Computer Science*, *28*(8).

Injadat, M., Moubayed, A., Nassif, A. B., & Shami, A. (2020). Multi-split optimized bagging ensemble model selection for multi-class educational data mining. *Applied Intelligence*, *50*, 4506-4528.

Mythili, M. S., & Shanavas, A. M. (2014). An Analysis of students' performance using classification algorithms. *IOSR Journal of Computer Engineering*, *16*(1), 63-69.

Ouyang, F., Wu, M., Zheng, L., Zhang, L., & Jiao, P. (2023). Integration of artificial intelligence performance prediction and learning analytics to improve student learning in online engineering course. *International Journal of Educational Technology in Higher Education*, *20*(1), 1-23.

Ramaswami, M., & Bhaskaran, R. (2009). A study on feature selection techniques in educational data mining. *arXiv preprint arXiv:0912.3924*.

Romero, C., & Ventura, S. (2020). Educational data mining and learning analytics: An updated survey. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, *10*(3), e1355.

Sabri, M., Zahid, M., Abd Majid, N. A., Hanawi, S. A., Talib, N. I. M., & Yatim, A. I. A. (2023). Prediction model based on continuous data for student performance using principal component analysis and support vector machine. *TEM Journal*, *12*(2).

Shetu, S. F., Saifuzzaman, M., Moon, N. N., Sultana, S., & Yousuf, R. (2021). Student's performance prediction using data mining technique depending on overall academic status and environmental attributes. *In International Conference on Innovative Computing and Communications: Proceedings of ICICC 2020, Volume 2* (pp. 757-769). Springer Singapore.

VeeraManickam, M. R. M., Mohanapriya, M., Pandey, B. K., Akhade, S., Kale, S. A., Patil, R., & Vigneshwar, M. (2019). Map-reduce framework based cluster architecture for academic student's performance prediction using cumulative dragonfly based neural network. *Cluster Computing*, *22*(Suppl 1), 1259-1275.

Xue, H., & Niu, Y. (2023). Multi-Output Based Hybrid Integrated Models for Student Performance Prediction. *Applied Sciences*, *13*(9), 5384.

Yağcı, M. (2022). Educational data mining: prediction of students' academic performance using machine learning algorithms. *Smart Learning Environment*, *9*(11). https://doi.org/10.1186/s40561-022-00192-z.

Yılmaz, N., & Sekeroglu, B. (2019, August). Student performance classification using artificial intelligence techniques. *In International Conference on Theory and Application of Soft Computing, Computing with Words and Perceptions* (pp. 596-603). Cham: Springer International Publishing.

Zaffar, M., Hashmani, M. A., Savita, K. S., & Rizvi, S. S. H. (2018). A study of feature selection algorithms for predicting students academic performance. *International Journal of Advanced Computer Science and Applications*, *9*(5).

Zhang, X., Liu, J., Zhang, C., Shao, D., & Cai, Z. (2023). Innovation Performance Prediction of University Student Teams Based on Bayesian Networks. *Sustainability*, *15*(3), 2335.