eMonitorMo: An Early Detection of Out-of-School Youths at Risk of Non-completion in a Non-formal Education Using Gradient Boosting Method

Jocelyn B. Barbosa[1*], Quinto A. Tan Jr.[1], Matthew R. Maulion[1]
Charlane B. Vallar[1], Jomar C. Llevado[1], Dario C. Miñoza[1]
and Maria Teresa M. Fajardo[2]
[1]College of Information Technology and Computing
[2]Division of Extension and Community Relations
University of Science and Technology of Southern Philippines – Cagayan de Oro
Cagayan de Oro City, 9000 Philippines
[*]*jocelyn.barbosa@ustp.edu.ph*

Date received: August 8, 2024
Revision accepted: December 15, 2024

Abstract

Out-of-School Youth (OSY) remains one of the major issues in many countries around the world. In the Philippines, approximately 3.8 million Filipino youths lacked access to formal education. Hence, the government has been implementing non-formal education programs to provide educational and career opportunities. However, low learner success rates have been prevalent, which may put the government's investment at risk of being wasted. Thus, the early detection of a learner's success is needed. There have been a number of studies conducted to predict students at risk of dropping, but none of them tackle out-of-school youth in non-formal education. In this study, the researchers introduced an innovative approach to mapping OSY through the development of a web and mobile-based system that includes early detection of OSYs at-risk of non-completion in ALS program using different classifiers. There were 17 potential independent variables identified in building the model, with "distance between the learning center and the learner's home" and "corresponding travel time" emerging as the top key predictors. The researchers considered decision tree, logistic regression, kNN, SVM, random forest, and gradient boosting as the most appropriate methods to build a model given a dataset size of 3,158. Experiments reveal that among the six methods employed, gradient boosting proved to be superior having an accuracy of 94.17%, with an area under curve (AUC) of 99.40%, thereby achieving better performance of at most 4.2% higher than the other 5 methods. This study aids administrators in identifying at-risk learners and delivering targeted interventions to boost completion rates.

Keywords: alternative learning system, gradient boosting method, non-formal education, out-of-school youth, predictive modelling

---

1. Introduction

Out-of-School Youth (OSY) remains one of the major issues in many countries around the world. A 2019 report by the United Nations Educational, Scientific and Cultural Organization (UNESCO) highlighted that with 258 million children, adolescents, and youth out of school, the global education crisis persists (UNESCO, 2019). In the Philippines alone, approximately 3.8 million Filipinos aged 6 to 24 years old lacked access to formal education making it one of the nine countries in Southeast Asia that was reported to have an increasing number of OSYs (Annual Poverty Indicators Survey, n.d.).

To address this pressing issue, a significant number of programs were established by the government to mitigate the issues on OSYs alongside with the intention to help them through educational, career and employment opportunities (Annual Poverty Indicators Survey, n.d.). This includes the implementation of the different non-formal education programs such as the Department of Education – Alternative Learning System (ALS), skills training, among others. However, despite the fact that the programs for OSYs across the region are generally supportive and enabling ones, many children and youths are still excluded from the formal system. This is due to the inefficient methods used to map out youth, particularly in rural areas. Additionally, low success rates among learners raise concerns that the government's investment may be wasted or result in inefficient budget allocation. Thus, recommendations have been put forth as possible ways forward. One of which is to strengthen and further the efforts in mapping and monitoring of OSYs (e.g., Technical Education and Skills Development Authority (TESDA) trainees, ALS learner's progress monitoring, etc.) and determine the likelihood of an OSY to succeed in the non-formal education program so as to plan for some interventions to take. Initial attempts for mapping and monitoring of OSYs were introduced.

However, significant number of drawbacks are identified, including (1) inconsistencies and/or duplication of records especially those who transferred residences, (2) high likelihood of inefficient mapping of OSYs especially those living in rural areas, (3) insufficient presentation or visualization of appropriate reports or data analytics for the management (e.g., DepEd ALS division, central office, etc.), which is normally done thru spreadsheets, and (4) absence of a predictive model to determine the likelihood of an ALS learner to complete the non-formal education program. Monitoring the completion rates of OSY in the ALS is essential. However, as of 2021, DepEd-

ALS Region 10 lacks an effective system to track whether OSY students complete the program after enrollment (R. Mahinay, personal communication, 2021). This gap in monitoring hinders government agencies from evaluating the efficiency of budget allocations, making it challenging to implement timely and appropriate interventions.

Many studies have explored academic performance prediction with data processing and outcome forecasting being central to these efforts (Qu *et al.*, 2018). Kostopoulos *et al.* (2019) used semi-supervised regression to predict final grades, while Yang *et al.* (2020) analyzed homework submissions to forecast outcomes based on procrastination. Nieto-Reyes *et al.* (2021) created a system that uses early assignment data to predict academic performance, helping teachers assess student success. On the other hand, Miao (2020) introduced an improved model using neural networks with support vector machines and principal component analysis. However, these models still face issues with data sources, accuracy, and efficiency. Other studies have investigated supervised machine learning for student profiling and success prediction. Devasia *et al.* (2016) identified Naïve Bayes as the most effective algorithm using demographic, academic, and extracurricular data. Febro and Barbosa (2017) utilized the Random Forest (RF) algorithm to detect potential university dropouts by modeling the transition from school to university. Additionally, Cardona and Cudney (2019) conducted a study that predicted student retention using SVM. Similarly, Liu *et al.* (2022) created a predictive model for the academic performances of college students utilizing the education system data. Alija *et al.* (2016) introduced a method in determining the key predictors that influence students' perception of course experience. While promising, these models need improvements in design and implementation, particularly in expanding data sources, enhancing prediction accuracy, and boosting efficiency. Moreover, all of them focused on higher education, and none of them tackle out-of-school youth in a non-formal education such as the alternative learning system (ALS) program implemented by the DepEd. Moreover, developing an OSY mapping system with predictive models for practical use is challenging and still underway.

In this study, the researchers introduced an innovative approach to mapping OSY through the development of a web and mobile-based system that includes early detection of OSYs at-risk of non-completion in ALS program using different classifiers. The study aimed to: (a) create a mobile-based application that performs profiling of OSYs who can be prospect learners/trainees under ALS/TESDA program; (b) develop a web-based system that allows registration of youths to different programs (e.g., DepEd ALS, TESDA, etc.); (c) build a predictive model that determines the likelihood of an OSY to successfully complete the said non-formal education program; and (d) test and evaluate the efficiency of the predictive model as integrated in the web-based eMonitorMo app.

2. Methodology

2.1 eMonitorMo Web and Mobile-based System Architecture

The system architecture illustrates the flow of user interactions with the platform, as depicted in Figure 1. It integrates multiple web technologies to ensure smooth access to data from both computers and smartphones. A key feature of the system is the automatic classification or early detection of ALS learners or OSYs at risk of non-completion in non-formal education. Additionally, the system includes a module that facilitates essential CRUD (Create, Read, Update, Delete) operations, enabling efficient data manipulation and management throughout the platform. This design ensures scalability and ease of use for various devices.

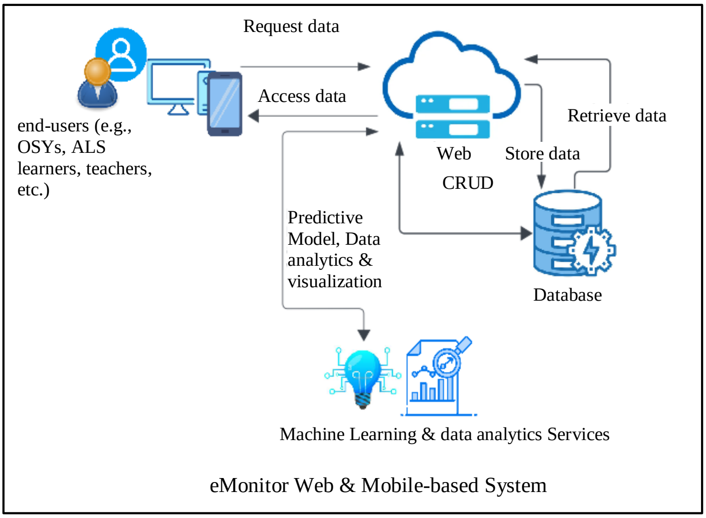


Figure 1. eMonitorMo System Architecture

2.2 Building the Predictive Model for OSY at Risk of Non-Completion in a Non-formal Education Classification

2.2.1 Framework of the Predictive Model for Integration to eMonitorMo System

The predictive modeling process includes these key steps: (1) Data Collection: aggregating data from diverse and reliable sources to ensure a robust dataset; (2) Data Preprocessing: cleaning, filtering, and transforming raw data into a suitable format for modeling by addressing issues like missing values, outliers, and irrelevant features; (3) Model Development: constructing a model using the pre-processed data; (4) Training: teaching the model to find patterns using a partition of the data; (5) Testing: evaluating the performance of the model on unseen data to evaluate generalizability; and (6) Results and Validation: analyzing performance metrics such as accuracy, precision, sensitivity, specificity, and F1-score, among others. Figure 2 outlines how the system processes the data and is able to generate prediction analysis on the ALS Learners Completion rate in the program.

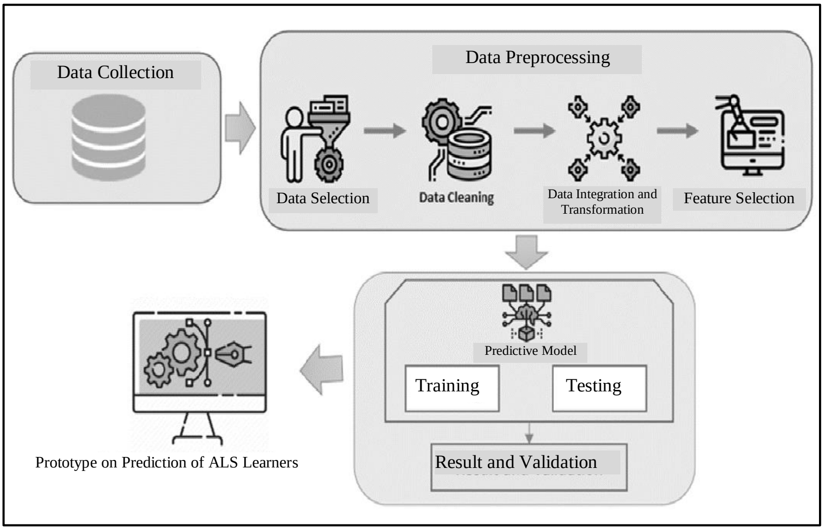


Figure 2. Framework of the predictive model for integration to eMonitorMo app

2.2.2 Creating the Predictive Model

Dataset and Predictors or Features

The dataset was obtained from ALS Region 10 between years 2017 to 2020 based on the standardized forms used by ALS teachers provided by the national level. Subjects are the ALS learners coming from different places in Region X. In the study conducted, a dataset of 3,158 records was used, each of which are pre-labeled by ALS teachers with completers (i.e., those who passed the accreditation and equivalency test) and non-completers (i.e., those who did not pass the accreditation and equivalency test) based on the historical records they gathered.

After pre-processing (e.g., removal of instances with missing values, etc.), the dataset was equally partitioned accordingly into two (2) classes with labels (i.e., 1,579 instances for “not at risk” of non-completion of the ALS program, which is equivalent to “completer” as originally pre-labelled, and 1,579 for “at risk” of non-completion or “non-completer”).

The data collected generally include the ALS Learners’ Basic Profile and Educational Information. However, from the various data fields available in the ALS form, 17 potential variables were identified as likely to intuitively influence the model's prediction of the target variable (i.e., determining whether an individual is 'at risk' or 'not at risk'). Table 1 presents a detailed description of independent and dependent/target variables. The 17 predictors or features were used in building a predictive model to detect OSYs at risk of non-completion in the ALS program, which involves training and testing the model. The method of which is discussed in the next subsection.

While the fields used may be similar to that of the other regions, the data itself may not necessarily represent the demographics of the other regions as there were no data collected from them yet. However, this does not hinder ALS to use the web and mobile-based system itself as the user interface allows other regions and that the predictive model may be applied and may work well as the model were trained via k-fold cross validation to avoid overfitting. Furthermore, the data acquired from other regions via the eMonitorMo system may be used to retrain the dataset in order to improve the model.

Table 1. List of predictors used in building the predictive model

Variables	Description	Values
AgeGroup	Age range	1 (10-19), 2 (20-29), 3 (30-39), 4 (40-50)
Gender	Gender	Male, Female
Civil Status	Civil Status	Single, Married, Widow, Separated, Common Law, Solo parent
Religion	Religion	Catholic, Non-Catholic
PWD	Person with disability	1 (Yes), 0 (No)
Mother Tongue	Mother Tongue	Cebuano, Tagalog, English, Visayan
FathersWork	Father's Occupation	Farmer, Fisherman, Driver, Self-employed, etc.
MothersWork	Mother's Occupation	Housekeeper, Housewife, Self-employed, Minimum earner, etc.
NumberOfParentsWorking	Number of ParentsWorking	0, 1, 2
LastGradeAttended	Last Grade Level Attended	01-Oct
NeedToWork	Dropped out from schools due to employment	1 (Yes), 0 (No)
Attended ALS before	Attended ALS Session before	1 (Yes), 0 (No)
CompletedALS	Completed ALS session before	1 (Yes), 0 (No)
Distance	Distance from learning center to home	0->100 (in Kms)
TravelTime	Time travelled from learning center home	0->1000 (in minutes)
ModeofTranspo	Mode of transportation from home to learning center	Walking, motorcycle, PUV, bicycle, etc.
NoOfHoursAvailable	Available number of hours to join ALS session	0->40 (hours)

Classification Model Formation for Predicting OSY at Risk of Non-completion in ALS Program

In this study, the researchers performed a training-and-testing activity from the 3,158 samples in the dataset. During this phase, the data is split into training, testing, and validation sets using k-fold cross validation. Each of the 6 models or classifiers is trained on k-1 folds and tested on the remaining fold. This process is repeated k times, with each fold serving as the test set, and the performance measures such as accuracy, precision, recall/sensitivity, specificity and F1-score are calculated in each fold. The results are then averaged to provide a more reliable estimate of the model performance. For example, given the dataset with 3,158 samples, $k = 10$ was chosen dividing it into 10 folds, each containing around 316 samples. A model (e.g., gradient boosting, random forest, etc.) is trained on 2,844 samples (i.e., 9 folds) and tested on the remaining 316 samples. This process is repeated 10 times, and the average performance is reported. This method helps reduce overfitting and provides a more robust evaluation of the model. This process is applied to each of the classification methods that were employed, namely: DT, logistic regression, support vector classifier, kNN, random forest, and gradient boosting method. Apart from the training-testing process, the researchers performed settings of hyperparameters and optimization techniques as shown in Table 2.

Table 2. Summary of the processes performed for each classification method employed in the dataset

Classification method	Training process	Hyperparameters	Optimization techniques
Decision Tree	Builds a single tree by recursively splitting data based on feature values.	Max depth, min samples to split/leaf, and the criterion for splitting.	Pruning to avoid overfitting and control complexity.
Logistic Regression	Models the probability of class membership using a logistic function, optimizing the model through maximum likelihood estimation.	Regularization strength (C) and solver type (e.g., liblinear, newton-cg).	Regularization techniques (L1, L2) to prevent overfitting and ensure model stability.
Support Vector Classifier	Finds the optimal hyperplane that maximizes the margin between classes by transforming data into a higher-dimensional space.	Kernel type, regularization parameter (C), and margin tolerance.	Uses kernel tricks, soft margin parameter, and hyperplane optimization to improve classification accuracy.
k-NN (k-Nearest Neighbors)	Classifies data based on the majority class of the nearest k neighbors in the feature space.	Number of neighbors (k) and distance metric (e.g., Euclidean or Manhattan).	No explicit training, but hyperparameters like *k* are optimized using cross-validation.
Random Forest	Creates multiple decision trees using bootstrap sampling and random feature selection, combining predictions to reduce overfitting.	Number of trees, max features, and max depth.	Bagging and random feature selection to improve robustness and reduce overfitting.
Gradient Boosting method	Trains models sequentially, where each tree corrects errors of the previous tree by minimizing a loss function, improving model performance over time.	Learning rate (*eta*), number of trees, max depth, and subsampling rate.	Shrinkage (learning rate), tree constraints, and early stopping to prevent overfitting, along with feature selection and subsampling to enhance generalization.

Gradient Boosting Method and How it is Used in this Study

Gradient Boosting is a robust boosting strategy that boosts the performance of weak learners by combining them into a strong learner. It is a machine learning technique that builds models sequentially by optimizing the residual errors of prior models (GeeksforGeeks, 2023). Figure 3 shows the gradient boosted trees for regression.

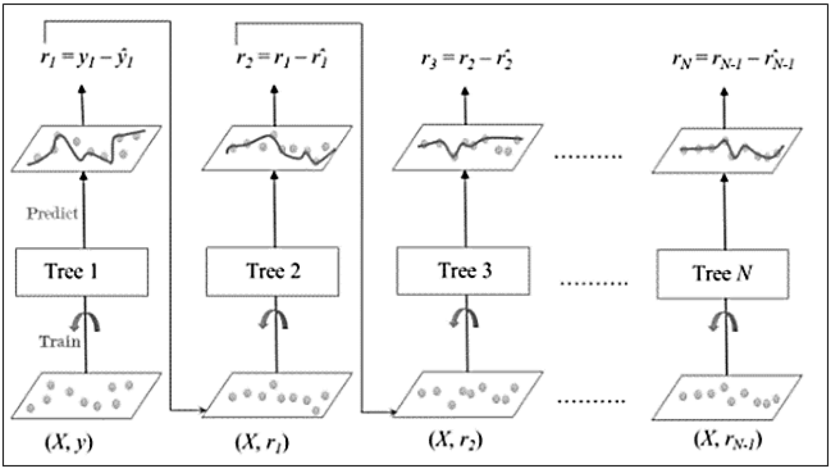


Figure 3. Gradient Boosted Trees for Regression (GeeksforGeeks, 2023)

In this study, the gradient boosting model was trained sequentially, with each weak learner correcting the previous model's errors using gradient descent. Key hyperparameters include the number of trees, learning rate, tree depth, and subsampling rates, which balance bias and variance. Regularization techniques like shrinkage, tree constraints, and early stopping help prevent overfitting, while subsampling and feature selection improve generalization.

By employing gradients to modify weights, Gradient Boosting is a reliable method that is less susceptible to outliers (GeeksforGeeks, 2023). It creates an ensemble of M trees, each of which fixes the mistakes in the preceding tree. The first tree is trained on the feature matrix (X) and labels (y), and subsequent trees are trained on the residual errors. "Shrinkage," a crucial parameter, scales the prediction of each tree according to a learning rate. For best results, a smaller eta or learning rate demands more trees. The final prediction is made by combining the outputs of all trees. This is represented in Equation 1.

$$y(pred) = y_1 + (eta \times r_1) + (eta \times r_2) + \cdots + (eta \times r_N) \qquad (1)$$

Below is a detailed explanation of the training process, hyperparameter settings, and optimization techniques used:

Sequential Training

The training process in Gradient Boosting involves building a series of weak learners (typically decision trees) in a step-by-step manner. Each new model aims to reduce the errors made by the ensemble of previously trained models. The key steps are: (1) Initialize the Model: start with a baseline prediction, often the mean (for regression) or log odds (for classification); (2) Compute Residuals: for each iteration, compute the residuals (errors) between the predicted values and the actual target values; (3) Fit Weak Learners: train a weak learner (e.g., a shallow decision tree) on the residuals; (4) Update Predictions: add the new learner's predictions to the ensemble using a scaling factor (learning rate); and (5) Iterate: repeat the process for a predefined number of iterations or until the residuals are sufficiently minimized.

Hyperparameter Settings

Proper tuning of hyperparameters is essential for the effectiveness of Gradient Boosting. The key hyperparameters and their typical settings are as follows: (1) Number of Trees (*n_estimators*), which determines how many boosting iterations are performed; (2) Higher values can improve performance but may lead to overfitting; (3) Typical range: 100–500; (4) Learning Rate (*learning_rate*); (5) Max Depth (*max_depth*); (6) Minimum Samples Split (*min_samples_split*) and Leaf (*min_samples_leaf*); (7) Subsample (*subsample*); and (8) Column Subsampling (*colsample_bytree*):

Optimization Techniques Used

Gradient Boosting incorporates several optimization strategies to improve performance and prevent overfitting. These include: gradient descent optimization, regularization techniques, randomness introduction, and feature importance handling. Gradient Boosting automatically assigns importance to features based on their contribution to reducing the loss function. This helps the model focus on the most relevant features.

3. Results and Discussion

3.1 Predictive Model Performance

Experiments reveal that gradient boosting method works best in predicting whether an ALS learner is at risk or not at-risk of non-completion in ALS program. Figure 4 presents a comparison of the average performance of the six (6) different classifiers employed in the dataset, namely: decision tree, logistic regression, kNN, SVM, random forest, and gradient boosting. Gradient boosting method achieves a better performance on harmonic mean or F1 score of 94.2% with a range of 1.2% to 21.6% higher than the other methods employed. Similarly, the gradient boosting method is superior among any other methods used in the experiment with an accuracy of 94.2% and precision of 94.4%. This also holds true in terms of recall, sensitivity, and specificity having approximately the same value of 94.2%. This further reveals that gradient boosting yields more stable results than the other classifiers used.

To facilitate better understanding of the results of the k-fold cross validation when gradient boosting was applied, the sample results in each iteration were provided for demonstration purposes (see Table 3). As presented in the table, if the average of each performance metric is computed, the results are the same with the ones presented in Table 4. For example, the sensitivity value of a classifier (gradient boosting) is 94.17%.

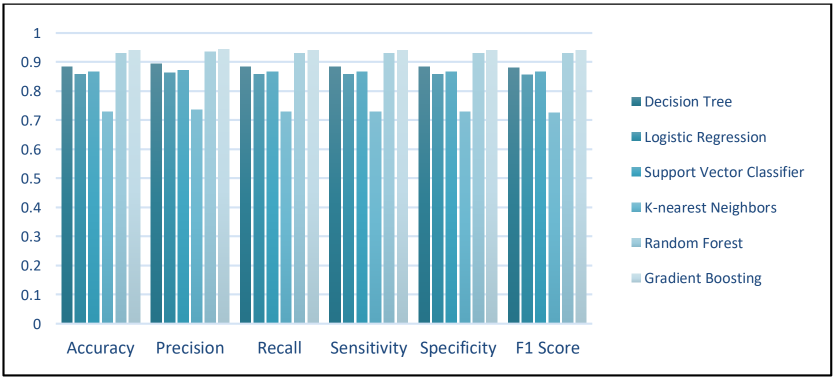


Figure 4. Comparison of the performance of the six methods for early prediction of OSY or learner at risk of non-completion in ALS program

Table 3. Sample results in each iteration when k-fold cross validation is applied to gradient boosting method for early prediction of the learners at risk of non-completion in ALS

No.	Fit time	Score time	Accuracy	Precision	f1_score	recall	Sensitivity	Specificity
0	0.1052	0.0020	0.8323	0.8382	0.8315	0.8323	0.8323	0.8323
1	0.1301	0.0051	0.9778	0.9788	0.9778	0.9778	0.9778	0.9778
2	0.1164	0.0020	0.9589	0.9620	0.9588	0.9589	0.9589	0.9589
3	0.0993	0.0041	0.9462	0.9477	0.9462	0.9462	0.9462	0.9462
4	0.0994	0.0021	0.9905	0.9907	0.9905	0.9905	0.9905	0.9905
5	0.1082	0.0029	0.9747	0.9759	0.9747	0.9747	0.9747	0.9747
6	0.1115	0.0030	0.9430	0.9489	0.9428	0.9430	0.9430	0.9430
7	0.1077	0.0030	0.9241	0.9241	0.9240	0.9241	0.9241	0.9241
8	0.1088	0.0030	0.9048	0.9080	0.9046	0.9048	0.9046	0.9046
9	0.1128	0.0020	0.9651	0.9660	0.9651	0.9651	0.9650	0.9650

A closer and detailed look at the classifier's performance, as displayed in Table 4, yields interesting facts regarding the specific capabilities of the six classification techniques. When it comes to separating students who are at risk of not completing the ALS program from those who are not, the gradient boosting method proves to be superior among other methods.

Table 4 presents a comparison of the average performance of the six different classifiers employed in the dataset: decision tree, logistic regression, kNN, SVM, random forest, and gradient boosting.

Table 4. Comparison of the performance of the six methods in tabular form for early prediction of OSY or learner at risk of non-completion in ALS

	Decision Tree	Logistic regression	Support vector classifier	K-nearest neighbors	Random forest	Gradient boosting	Best score
Accuracy	0.8832	0.8578	0.8664	0.7277	0.9300	0.9417	Gradient Boosting
Precision	0.8933	0.8635	0.8706	0.7351	0.9361	0.9440	Gradient Boosting
Recall	0.8832	0.8578	0.8664	0.7277	0.9300	0.9417	Gradient Boosting
Sensitivity	0.8832	0.8578	0.8664	0.7277	0.9300	0.9417	Gradient Boosting
Specificity	0.8832	0.8578	0.8664	0.7277	0.9300	0.9417	Gradient Boosting
F1 Score	0.8795	0.8572	0.8660	0.7257	0.9295	0.9416	Gradient Boosting

For determining whether or not an ALS learner is at risk of non-completion in ALS program, gradient boosting method achieves a better performance on sensitivity of 94.17%, with a range of 1.17% to 21.4% higher than the other methods employed. While the random forest classification method ranks second that also shows good results comparable with the gradient boosting method, the researchers are more concerned on designing a classifier that yields stable results on sensitivity performance measure without necessarily sacrificing the specificity or fallout or the probability of false alarm. This means that the primary focus is on stabilizing sensitivity, as it is aimed at minimizing false negatives (i.e., reducing them to near zero) in order to maximize true positives, thereby achieving a sensitivity value close to 100%. Minimizing false negatives means ensuring that when the predictive model indicates a negative result (i.e., a learner is not at risk of non-completion), the actual outcome or unseen instance indeed confirms this, showing the learner is a completer of the program.

Additionally, the empirical results reveal that the gradient boosting method is superior among any other methods tested in terms of other performance metrics with an accuracy, precision, recall, and F1-score of 94.17%, 94.40%, 94.17% and 94.16%, respectively. Hence, the researchers chose gradient boosting method to be integrated in the developed eMonitorMo web-based system for the ALS teachers and administrators to see whether or not the newly enrolled student, after filling-out the forms, is at-risk or not at-risk of non-completion in ALS program.

To illustrate the diagnostic ability of the gradient boosting method-based classifier, a graphical plot called receiver operating characteristic (ROC) curve is created (see Figure 5). ROC shows the graphical plot while varying the discrimination threshold. The ROC curve is a plot of the true positive rate (TPR) on the y-axis versus the False Positive Rate (FPR) on the x-axis for every possible classification threshold. The proposed predictive model using gradient boosting method achieves better performance in AUC of 0.5%-4.2% higher than other five methods in discriminating “at risk” from “not risk” OSY learners. Experiments reveal that the predictive model introduced in this study based on the gradient boosting method yields more stable results.

On the other hand, Table 5 presents the importance of each feature used relative to the performance of the predictive model based on the gradient boosting method.

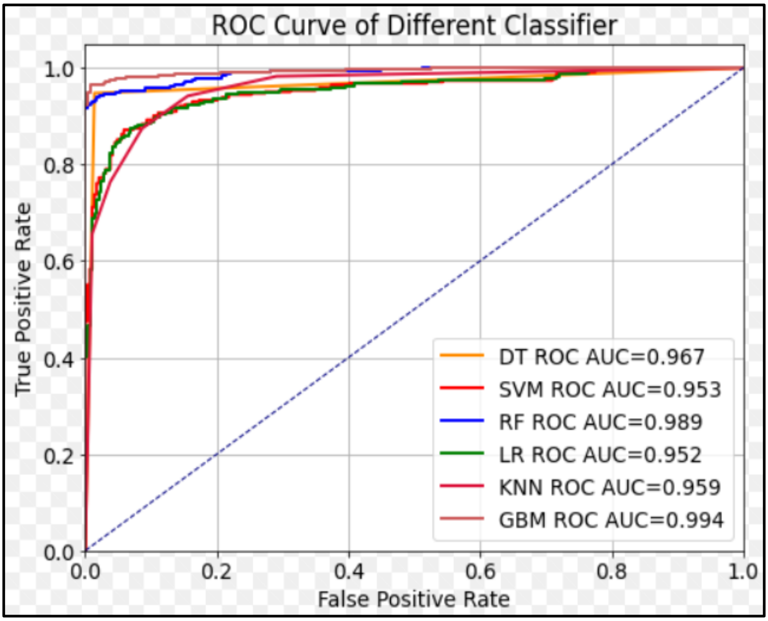


Figure 5. Receiver operating curve

Table 5. Feature importance of the created model based on the gradient boosting method

No.	Factors	Percentage of importance
1	Distance from community learning center (CLC) to home	49.66%
2	Travel time to go to the CLC	18.59%
3	Employment	5.80%
4	Available time to attend	5.24%
5	Ethnic group	3.87%
6	Mother tongue	3.77%
7	Mode of transportation going to the nearest CLC	3.71%
8	Attended ALS session before	1.82%
9	Number of parents working	1.52%

The "distance from community learning center to home" contributed the highest to the model's predictive accuracy, accounting for approximately 49.66% of the total feature importance score. "Travel Time to go to the CLC" ranked second in importance, with a contribution of 18.59%, providing significant discrimination between target classes – "at risk" or "not at risk". As for the "employment and available time to attend" features, although their individual importance scores are lower, they complement other features, particularly when combined with the rest of the five features presented in rows 5 to 9 in Table 3.

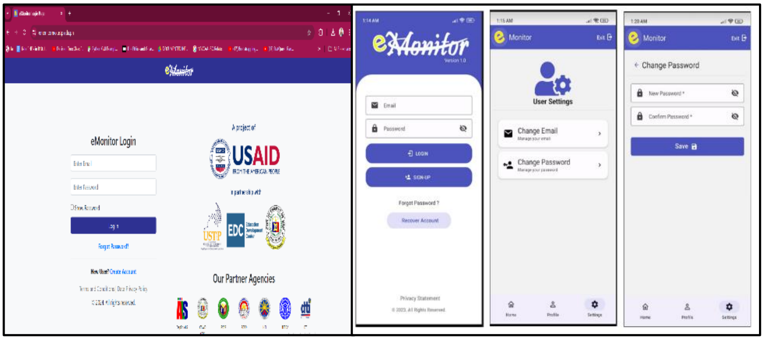
3.2 Web and Mobile-based Performance

3.2.1 eMonitorMo App User Interface

Following the system architecture and design presented in Figure 1, a web and mobile-based application was developed for the out-of-school youth, and ALS teachers and administrators. Figure 6 presents the sample screen shots of the eMonitorMo web and mobile-based app. The login page allows users to enter their credentials to access the system, with options for "Forgot Password", "change password", and registration. New users can sign up by providing basic details like username/email and password. Once logged in or registered, users are directed to the dashboard. The dashboard is the main interface users see after logging in, providing an overview of key information of the out-of-school youths, ALS teachers and other key agencies. It is designed for simplicity and quick navigation, helping users manage tasks efficiently.

Figure 7 shows a sample screenshot of a data visualization chart showing the number of youths registered in the eMonitorMo. It provides a clear, graphical representation of registration trends over time, allowing users to easily check registered users, which includes filters to view data by specific regions, province and city.

On the other hand, Figure 8 presents the form duly filled-out by the out-of-school enrolled in ALS program. As a result of the integration of the gradient boosting method-based predictive model, the ALS teacher or administrator is able to see upon enrollment if a learner is at risk or not at risk of non-completion in ALS program (i.e., the target variable class = 1 or 0, for at risk and not risk, respectively).



(a) (b)

Figure 6. Sample screenshots of the eMonitorMo web and mobile-based app: Web-based app landing page(a); and mobile-based app registration and login module (b)

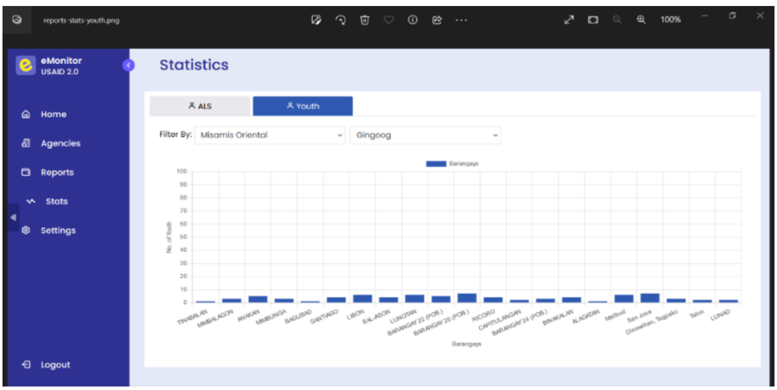


Figure 7. Sample screen shot of data visualization

3.2.2 System's Usability Test

To test the system's usability, a survey was conducted using the system's usability scale (SUS) test instrument (Brooke, 2013). There were 20 respondents involved in the survey, which include the ALS learners, teachers and implementers. The results of the survey are summarized in Tables 6 and 7. To facilitate clear understanding and interpretation, the survey questions were categorized into odd-numbered questions (positively framed statements) and even-numbered questions (negatively framed statements). These two categories are presented separately for better clarity and understanding.

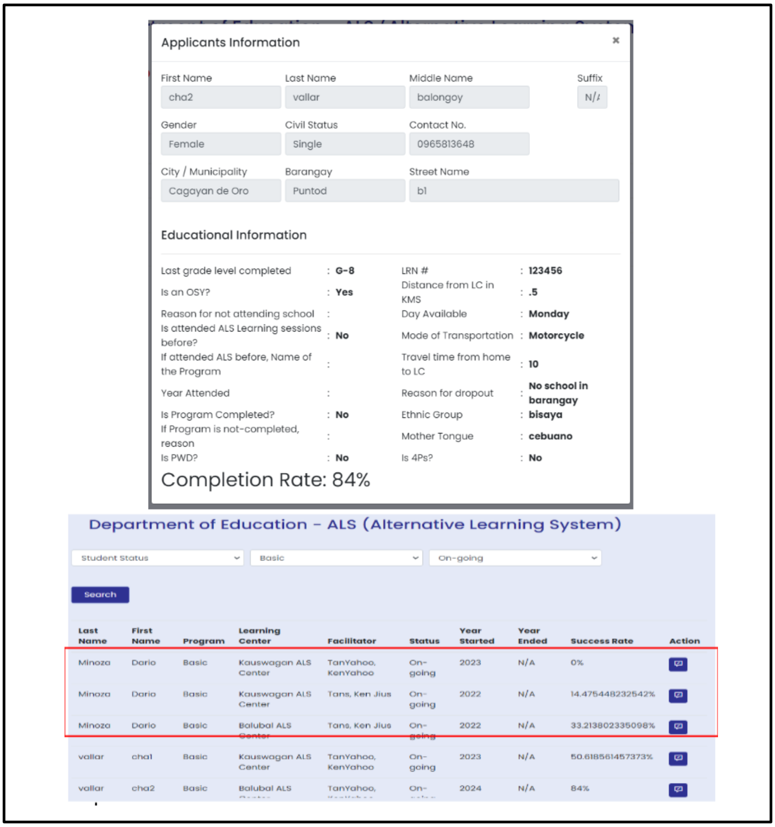


Figure 8. User Interface (UI) samples after integrating the predictive model to the eMonitorMo app

As shown in Table 6, 95% of respondents (55% strongly agree and 40% agree) expressed a positive desire to use the system, while only 5% rated it as fair. Regarding ease of use, 80% of respondents (60% strongly agree and 20% agree) found the system easy to use, with 20% rating it as fair, indicating strong usability perceptions. A majority (85% of respondents [45% strongly agree and 40% agree]) felt the system's functions were well integrated, while 15% rated it as fair, reflecting positive views on functional integration. Additionally, 75% of respondents (35% strongly agree and 40% agree) believed the system could be learned quickly, highlighting its perceived learnability, though 20% rated it as fair and 5% disagreed. Finally, 80% of respondents (45% felt very confident and 35% felt confident) reported confidence in using the system, with 15% rating it as fair. Despite a small proportion (5%) who disagreed, these findings collectively suggest a positive perception of the system's usability, integration, and learnability.

Table 6. Responses to the positive statements of the system usability scale (SUS) test

Statements	5 Strongly Agree		4 Agree		3 Fair		2 Disagree		1 Strongly Disagree	
	f	%	f	%	f	%	f	%	f	%
1. I think that I would like to use this system.	11	55%	8	40%	1	5%	-	-	-	-
3. I thought the system was easy to use.	12	60%	4	20%	4	20%	-	-	-	-
5. I found the various functions in this system were well integrated.	9	45%	8	40%	3	15%	-	-	-	-
7. I would imagine that most people would learn to use this system very quickly.	7	35%	8	40%	4	20%	1	5%	-	-
9. I felt very confident using the system.	9	45%	7	35%	3	15%	1	5%	-	-

Table 7 presents the descriptive statistics for the even-numbered, negatively phrased statements in the SUS, highlighting user perceptions and potential challenges. The results indicate that 60% of respondents (25% strongly disagree, 35% disagree) did not find the system unnecessarily complex, reflecting its user-friendly design. Additionally, 65% expressed confidence in using the system independently, 80% felt it was consistent, and 85% found it manageable. Furthermore, 60% indicated they could use the system effectively, likely benefiting from the training provided to stakeholders. Overall, the findings suggest positive user perceptions of the system's usability, user-friendliness, and learnability, with opportunities for refinement to enhance the user experience further.

In assessing the system's overall usability, the odd-numbered questions were recalibrated by subtracting 1 from their scale position, and the total sum was multiplied by 2.5 to calculate the SUS score, which ranges from 0 to 100. Accordingly, an average SUS score is 68. Systems scoring above 68 are generally perceived as having above-average usability, while those scoring below 68 may indicate usability concerns (Brooke, 2013). The results of the eMonitorMo usability survey revealed an overall SUS score of 78, indicating above-average usability for the system. While this score reflects positive user reception and system acceptance, it also highlights opportunities for further refinement. These areas for improvement were explored through qualitative analysis to better understand how the system's usability could be elevated further.

Table 7. Responses to the negative statements of the SUS test

Statements	5 Strongly Agree		4 Agree		3 Fair		2 Disagree		1 Strongly Disagree	
	f	%	f	%	f	%	f	%	f	%
2. I found the system unnecessarily complex.	1	5%	2	10%	5	25%	7	35%	5	25%
4. I think I would need the support of a technical person to be able to use this system.	-	-	1	5%	6	30%	10	50%	3	15%
6. I thought there was too much inconsistency in this system.	-	-	1	5%	3	15%	11	55%	5	25%
8. I found the system very cumbersome to use.	-	-	-	-	3	15%	7	35%	10	50%
10. I needed to learn a lot of things before I could get going with the system.	1	5%	2	10%	5	25%	8	40%	4	20%

4. Conclusion and Recommendation

In this study, the researchers presented an innovative approach to effectively map and profile OSYs in the region. Furthermore, this study has several merits that are essential to real application that include: (a) building of a predictive model that determines the likelihood of an OSY to be at risk of non-completion in the ALS program, thereby providing early warning and prompt execution of interventions for the learner’s improvement; (b) design and development of mobile-based application that performs profiling of OSYs who can be prospect learners/trainees under ALS/TESDA program with positive overall sentiments of users; and (c) development of a web-based system that allows registration of youths to different programs (e.g., DepEd ALS, TESDA, etc.) with data visualization and integration of a predictive model having above average usability as rated by the users.

Furthermore, the "distance from community learning center to home" and "travel time to go to the CLC" were found to be the top key predictors and are strong indicators in discriminating learners from being "at risk" of non-completion in ALS program or "not at risk". This aids administrators and implementers in identifying at-risk learners and delivering targeted interventions, thereby boosting completion rate of ALS learners, which in turn make the government funds not to go in vain.

While the results of the study have several merits, the following are recommended to be considered for future research work. These include: (a) identification of specific program interventions. This further safeguards the government's investment for non-formal education. Identifying specific interventions for ALS implementers to take for those who are at risk of non-completion in ALS program may help improve the study; (b) improvement of user experience and system efficiency. While the overall sentiments are positive and suggest that the system is generally well-received, the results also point to specific areas where improvements could be made to enhance user experience and system efficiency further; and (3) employment of active learning models. Exploring other techniques like active learning or transfer learning model may improve the performance of the classifier, and combining the learner's formative assessment scores to the list of predictors may further improve the results of the study.

5. Acknowledgement

This work was supported by the USAID Opportunity 2.0 Grant No. 2021-0053-25 and University of Science and Technology of Science and Technology – Cagayan de Oro.

6. References

Alija, S., Snopce, H., & Aliu, A. (2016). Logistic regression for determining factors influencing students' perception of course experience. The Eurasia Proceedings of Educational and Social Sciences, 5, 99-106.

Philippine Statistics Authority. (n.d.). Annual poverty indicators survey (APIS). Retrieved from https://psa.gov.ph/content/annual-poverty-indicators-survey-apis

Brooke, J. (2013). SUS: A retrospective. Journal of Usability Studies, 8(2), 29-40.

Cardona, T., & Cudney, E.A. (2019). Predicting student retention using support vector machines. Procedia Manufacturing, 39, 1827-1833.

Devasia, T., Vinushree, T., & Vinayak, H. (2016). Prediction of students' performance using educational data mining. Proceedings of the 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), Ernakulam, India, 91-95.

Febro, J., & Barbosa, J. (2017). Mining student at risk in higher education using predictive models. Journal of Advances in Technology and Engineering Research, 3(4), 117-132.

GeeksforGeeks. (2023). Gradient boosting in ML. Retrieved from https://www.geeks forgeeks.org/ml-gradient-boosting/

Kostopoulos, G., Kotsiantis, S., Fazakis, N., Koutsonikos, G., & Pierrakeas, C. (2019). A semi-supervised regression algorithm for grade prediction of students in distance learning courses. International Journal on Artificial Intelligence Tools, 28, 1940001.

Liu, C., Wang, H., & Yuan, Z. (2022). A method for predicting the academic performances of college students based on education system data. Mathematics (Basel), 10(20), 3737. https://doi.org/10.3390/math10203737

Miao, J. (2020). A hybrid model for student grade prediction using support vector machine and neural network. Journal of Intelligent and Fuzzy Systems, 40, 1-11.

Nieto-Reyes, A., Duque, R., & Francisci, G. (2021). A method to automate the prediction of student academic performance from early stages of the course. Mathematics, 9(21), 2677. https://doi.org/10.3390/math9212677

Qu, S., Li, K., Zhang, S., & Wang, Y. (2018). Predicting achievement of students in smart campus. IEEE Access, 6, 60264-60273.

United Nations Educational, Scientific and Cultural Organization (UNESCO). (2019). New methodology shows that 258 million children, adolescents and youth are out of school. Retrieved from https://reliefweb.int/report/world/new-methodology-shows-258-million-children-adolescents-and-youth-are-out-school

Yang, Y., Hooshyar, D., Pedaste, M., Wang, M., Huang, Y., & Lim, H. (2020). Predicting course achievement of university students based on their procrastination behaviour on Moodle. Soft Computing, 24(24), 18777-18793.