

Predictive AI for Enhancing Clinical Pathways in Children with Learning Disabilities using XGBoost

Sujitha V.¹, Kishore P.², Nevin Nehal R.S.³, Afzal Khan M.⁴

¹Head of the Department, ^{2,3,4}Student, Department of Computer Science and Engineering, NPR College of Engineering and Technology, Natham, Dindigul, India.

E-mail: ¹sujithav@nprcolleges.org, ²p.kishore190805@gmail.com, ³nevinnehal786@gmail.com,

⁴Afzalkhanm1610@gmail.com

Abstract

Dyslexia screening in children involves computing techniques to analyse the learning patterns from the structured data generated by behaviour and language assessments. This research work proposes a machine learning technique to classify dyslexic cases by implementing Random Forest and XGBoost on a publicly available dataset consisting of 3,644 observations and 196 features extracted from demographics and language tasks. Steps in data pre-processing involves the missing observations handling, categorical data encoding, non-predictive columns elimination, and stratified splitting of training and testing datasets. Hyperparameter tuning is performed via stratified cross-validation, followed by independent testing of both ensemble models. Random Forest attained 95.1% accuracy with 94.3% precision, 93.8% recall, 94.0% F1-Score, and 0.97 ROC-AUC score, whereas XGBoost gave an accuracy of 91.8% and ROC-AUC of 0.94. Flask-based prediction model was created for providing individualized screening predictions. The platform shows how ensemble learning could be used in computational dyslexia screening and serves as an illustrative example of the decision support tool for preliminary screening.

Keywords: Classification, Dyslexia Screening, Random Forest, XGBoost, Feature Representation, Ensemble Learning, Binary Classification.

1. Introduction

Specific Learning Disabilities (SLD) involve persistent challenges in acquiring and applying academic and language-based skills. Dyslexia is a learning disability, which has its primary difficulty in reading and language processing abilities; these difficulties can differ significantly in terms of their linguistic and cognitive features. It means that the computational detection of dyslexia must consider the multiple features of the participants and performance of tasks as opposed to a single feature. The ML approach is appropriate for learning nonlinear relationships between structured variables and diagnosing classes [1], [3].

ML applications for Dyslexia-based classification have been considered based on behavioral, physiological, eye movement, and phonological features [4], [5], [9]. Among all the aforementioned feature extraction methodologies, structured assessment data offers a way to directly represent characteristics and behavior of the participants during linguistic exercises. The public dataset used in this work consists of 3,644 participants with 196 attributes based on demographics and 32 linguistic exercises [11].

In this research, random forest (RF) and extreme gradient boost (XGB) algorithms are used for building ensemble classifiers. RF uses bagging method to aggregate multiple predictions generated by multiple decision trees, while XGB builds a series of trees to learn an optimized ensemble by gradient boosting [2]. The proposed architecture includes data pre-processing, feature transformation, stratified partitioning, and hyperparameters optimization using cross-validation prior to model training. Trained models are then analyzed through performance evaluation metrics and confusion matrix before being implemented in a Flask application for predicting individual participants.

2. Related Works

Machine Learning (ML) based methods have been extensively explored to identify learning disability through behavior, physiology, and academic features. Current research demonstrates that the various forms of learning disabilities can be modeled computationally and classified through supervised learning algorithms. A machine learning framework for learning disability analysis has provided the evidence of effectiveness of predictive models for automated identification, laying down the initial foundation of computational diagnosis of

learning disabilities [1]. Classification of children with learning disabilities through EEG has also provided the evidence of utilization of neural network models in the classification task [3].

Dyslexia-focused research has explored eye-movement and gaze-related characteristics as discriminative signals. Machine-learning models have been applied to fixation and saccadic eye-movement features for predicting dyslexia severity, demonstrating that oculomotor characteristics can support severity-oriented classification [4]. A holistic eye-tracking approach has also been investigated for dyslexia detection, extending the use of gaze behavior beyond individual eye-movement parameters [5]. These approaches provide valuable evidence for automated screening but require specialized sensing or acquisition procedures that may limit their applicability in ordinary educational environments.

Other researches have investigated alternative computational modeling of reading disabilities and linguistic problems. Intelligent dyslexia diagnosis was accomplished by machine learning using fuzzy representation, and eye movement time series properties were utilized to identify dyslexic children with high risk [6], [7]. The possibility of reinforcement learning for diagnosing reading disabilities using eye-gaze analysis has further been explored showing the potentiality of adaptive learning in learning disabilities diagnostics [8]. Phonological features were further studied using machine learning methods in case of developmental dyslexia [9].

Recent research has taken the field of machine learning into the area of early detection of learning problems in educational systems. A tri-tier machine-learning approach has been introduced for the early detection of learning problems within the context of the basic school systems, focusing on structured screening in early phases of the educational system [10]. Ensemble learning techniques are suitable foundations for structured classification due to their boosting abilities [2].

2.1 Research Gap

Existing studies have demonstrated the application of ML techniques for learning disabilities and dyslexia classification in terms of behavioral, physiological, eye-tracking, and phonological features [1], [3]-[9]. Nevertheless, the application of structured individual-level assessment data including demographic features and linguistic task performance features with the help of tree-based ensemble classifier is understudied. Moreover, various classifiers are

often compared within various data representation and experiment settings. Hence, in this work, we introduce a unified classification setting based on Random Forest and XGBoost classifiers with equal pre-processing and parameter tuning for binary dyslexia classification.

3. Proposed Methodology

3.1 System Architecture

The proposed framework for dyslexia classification includes a sequence of stages such as dataset collection, data pre-processing, feature representation, data stratification, ensemble model creation, hyperparameters tuning, and performance evaluation. Firstly, the dyslexia dataset is collected and represented in a form of features. Then, pre-processing stage is used in order to take care of categorical data, missing values, and non-predictive fields. The pre-processed data set is further split into training and testing subsets by means of stratified sampling. The training subset is utilized for developing the Random Forest and XGBoost classifiers, whereas the hyperparameters optimization of these models is done via cross-validation. Further, the tuned models are retrained on the entire training subset and tested on the separate testing subset. The models produced are implemented in the user interface to make Dyslexia/Non-Dyslexia predictions for participants.

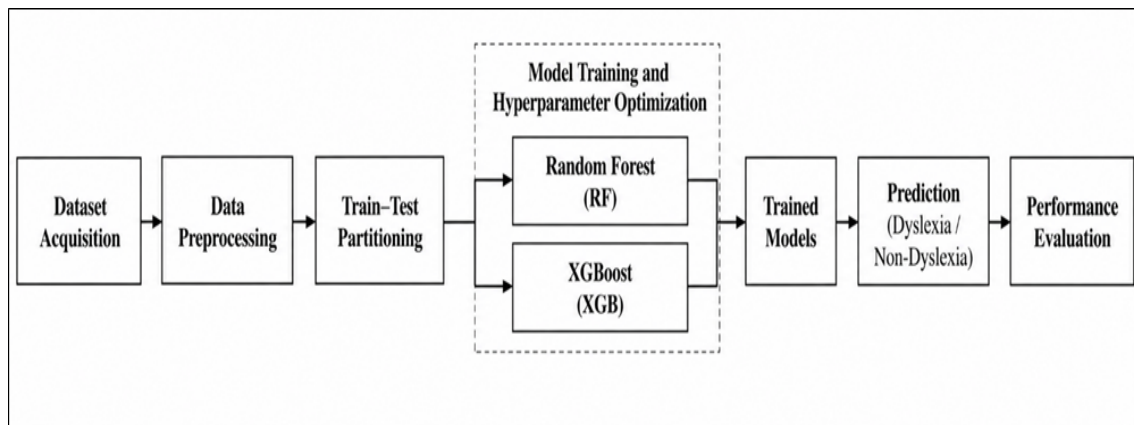


Figure 1. Proposed Dyslexia classification framework

3.2 Dataset Construction and Feature Representation

A publicly available dyslexia testing dataset was used for the development and testing of the proposed classification system [11]. It consists of 3,644 records of participants acquired via an online game-based test that was designed to measure characteristics of behavior and

language associated with dyslexia. Participants were children and teenagers between 7-17 years old, and there are 196 attributes in the dataset which were collected on the basis of demographic information and results of 32 language tests [11].

The predictor variables utilized in modeling were arranged in a feature vector expressed as

$$X_i = [x_{i1}, x_{i2}, \dots, x_{ip}] \quad (1)$$

where X_i denotes the feature vector of the i -th participant, x_{ij} represents the j -th retained predictor, and p denotes the number of predictor variables retained for model development after preprocessing. The diagnostic category for each individual participant was used as the binary response variable, whereby the two categories were Dyslexia and Non-Dyslexia. Below is a summary of the feature sets together with their characteristics as indicated in Table 1.

Table 1. Feature representation of the Dyslexia dataset

Feature Category	Description
Demographic Features	Age, gender, native language, and language-subject history
Interaction Features	Clicks, hits, and misses recorded during linguistic tasks
Performance Features	Score, accuracy, and miss rate obtained from the assessment tasks
Linguistic Features	Measures derived from the 32 linguistic exercises
Task-Level Features	Participant responses and performance characteristics across individual exercises
Diagnostic Target	Binary classification label representing Dyslexia or Non-Dyslexia

3.3 Data Preprocessing and Partitioning

The dataset was subjected to preprocessing in order to obtain a uniform numerical representation for developing the model. The steps involved in preprocessing were: identification of missing values, encoding of categorical features, elimination of irrelevant

features, and preparing the dependent variable for diagnosis using binary classification. The numeric features were left as they were, but the categorical features were converted into numeric features. After preprocessing, the entire dataset was split into training and testing datasets through stratified sampling in an 80:20 split. This implies that 2,915 observations were assigned to the training dataset and 729 observations to the independent testing dataset while maintaining the class distribution in the two splits. The training dataset was utilized to develop and tune the model, while the testing dataset was held out and used solely for performance evaluation purposes. The dataset split can be expressed as

$$D = D_{\text{train}} \cup D_{\text{test}}, |D_{\text{train}}| = 0.8|D|, |D_{\text{test}}| = 0.2|D| \quad (2)$$

where D denotes the complete dataset, while D_{train} and D_{test} represent the training and testing subsets, respectively.

3.4 Ensemble Classification

The two tree-based ensemble classifier methods considered are the Random Forest (RF) and the Extreme Gradient Boosting (XGB). These two classifiers were tested using the same training subset that had been preprocessed, under the same experiment conditions to facilitate comparison. The inputs fed to these two classifiers were the predictor variables, and the output was the label.

1. Random Forest

The Random Forest classifier was implemented using the technique of bootstrap aggregation where several decision trees were formed using different sets of samples of the training data. In every split, a random sample of the predictor variables was considered to form an optimal split. Majority vote among the individual trees formed the final output, which could be expressed as

$$\hat{y}_i = \text{mode} \{h_1(X_i), h_2(X_i), \dots, h_T(X_i)\} \quad (3)$$

where X_i denotes the feature vector of the i -th participant, h_t represents the prediction of the t -th decision tree, and T denotes the total number of trees.

2. XGBoost

XGBoost was used as a gradient boosting classifier where decision trees were built

consecutively to reduce the prediction residual error of the previous set of trees. While in the case of Random Forest independent tree building is used, XGBoost uses an iterative approach for building the ensemble by adding new trees to improve the objective function. The prediction can be represented as

$$\hat{y}_i = \sigma \left(\sum_{t=1}^T f_t(X_i) \right) \quad (4)$$

where f_t denotes the prediction function of the t -th boosting tree, T represents the number of boosting iterations, and $\sigma(\cdot)$ denotes the logistic transformation used to obtain the binary classification probability.

For both classifiers, the process involves assigning a probability value corresponding to each record input based on the learned patterns from the training data set. The predicted probability is compared with a classification threshold to assign the final class. All those records that satisfy the decision rule are labeled Dyslexia, and others will be assigned a class label of Non-Dyslexia.

3.5 Hyperparameter Optimization

Optimization of the hyperparameters was done in order to find good settings for the Random Forest and XGBoost models. This was done only using the training set in order not to introduce any leaks from the independent test set. The hyperparameters were evaluated using stratified cross-validation on the training set, ensuring that the class distribution is the same in all the folds. In Random Forest, the following optimization parameters were considered: number of trees, maximum tree depth, and approach to feature-subset selection. In XGBoost, optimization parameters included the number of boosting estimators, learning rate, maximum tree depth, and subsampling ratio. The combination of settings that offered the best validation results was chosen for further model training. The chosen models were trained on the full training set, while the independent testing subset remained untouched until final evaluation.

3.6 Predictive Evaluation

Evaluations of the performance of the two trained models of Random Forest classifier and XGBoost classifier were carried out by using the testing data that was set aside and did not participate in training and tuning of hyperparameters. Measures of Accuracy, Precision, Recall, F1-score, and ROC-AUC were selected in order to provide a complementing analysis of the

performance of binary dyslexia classification.

Represents the ratio of participants correctly identified to the entire test sample and is calculated as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

where TP , TN , FP , and FN denote true-positive, true-negative, false-positive, and false-negative predictions, respectively. Precision and recall measures were used to measure the correctness and sensitivity of positive class prediction, respectively, while the F1-score is their harmonic mean. In addition, the ROC-AUC metric was used to analyze the discriminative power of the classifiers at various classification thresholds. Confusion matrix analysis was used to pinpoint errors in specific classes and to understand the distribution of false positives and negatives.

3.7 End-to-End Screening Workflow

The proposed framework operates using a sequential processing pipeline where participant-based assessment records are mapped into binary predictions about dyslexia. Firstly, data preparation takes place, where all demographic and linguistic performance characteristics are arranged in an input representation format. In the second step, data preprocessing takes care of missing data, mapping categorical data, and elimination of uninformative features.

The processed data set is then divided into training and testing sets through stratified sampling in order to maintain the distribution of the classes. The training set is employed for the creation of the Random Forest and XGBoost classifiers, whereas hyperparameter tuning is carried out in the training set in order to decide on an appropriate model architecture without taking any input from the independent testing subset.

For a new participant record, the same pre-processing representation is generated, and then fed to the trained classifiers. All the classifiers predict the probabilities of occurrence of the two classes and the predicted probabilities are converted into binary form either indicating Dyslexia or Non-Dyslexia. Independent test set will then be used to find out the accuracy of the model and compare the predictive behavior of both classifiers. The trained classifier may then be accessed via the Flask application to provide participant-level predictions.

3.8 Implementation Environment

The proposed architecture was developed using Python as the main programming language. The datasets were handled and prepared using Pandas and NumPy libraries. Categorical data was preprocessed using Scikit-learn library along with train-test splitting in stratified manner, Random Forest classification, and evaluation. The XGBoost library was utilized for the construction of gradient-boosted tree classifier.

The deployment was organized as a modular pipeline consisting of data preparation, preprocessing, building of models, tuning of hyperparameters, predictions, and evaluation of the performance of the models. Both the ensemble models received the preprocessed participants' records with the same training and testing subsets. Stratified cross-validation was done during the tuning of the parameters, after which the chosen configuration was further trained with the entire training subset. The trained models produced a binary output representing Dyslexia and Non-Dyslexia per record.

For implementation purposes, a web application built using Flask was combined with the trained classifiers to make screening predictions at an individual level via the web interface. The application takes the necessary input attributes, processes them using the learned representation method, and then feeds the processed record to the classifier for prediction.

4. Results and Discussion

4.1 Comparative Classification Performance

The predictive performance of the Random Forest and XGBoost models was analyzed on the independent test data set based on the selected evaluation measures. The results of comparison are given in Table 2. Random Forest demonstrated stronger overall performance, whereas XGBoost produced slightly lower values across most measures.

Table 2. Comparative classification performance of the proposed models

Evaluation Measure	Random Forest	XGBoost
Accuracy	95.1%	91.8%
Precision	94.3%	90.7%

Recall	93.8%	92.1%
F1-score	94.0%	91.4%
ROC-AUC	0.97	0.94

The Random Forest model performs well, and its outcomes showed successful classification outcomes. XGBoost also provided reliable classification, although its overall performance remained comparatively lower. The relatively close recall values indicate that both models were effective in identifying dyslexia-related instances, while the higher precision and F1-score obtained by Random Forest indicate more consistent class prediction. The recall values that are similar in magnitude show that both models were able to detect instances associated with dyslexia. However, Random Forest outperformed XGBoost in terms of precision and F1-score measures. Moreover, the ROC-AUC values demonstrate high class separation performance by both models. Therefore, Random Forest was selected as the effective classifier in this research.

4.2 Class-Level Prediction Analysis

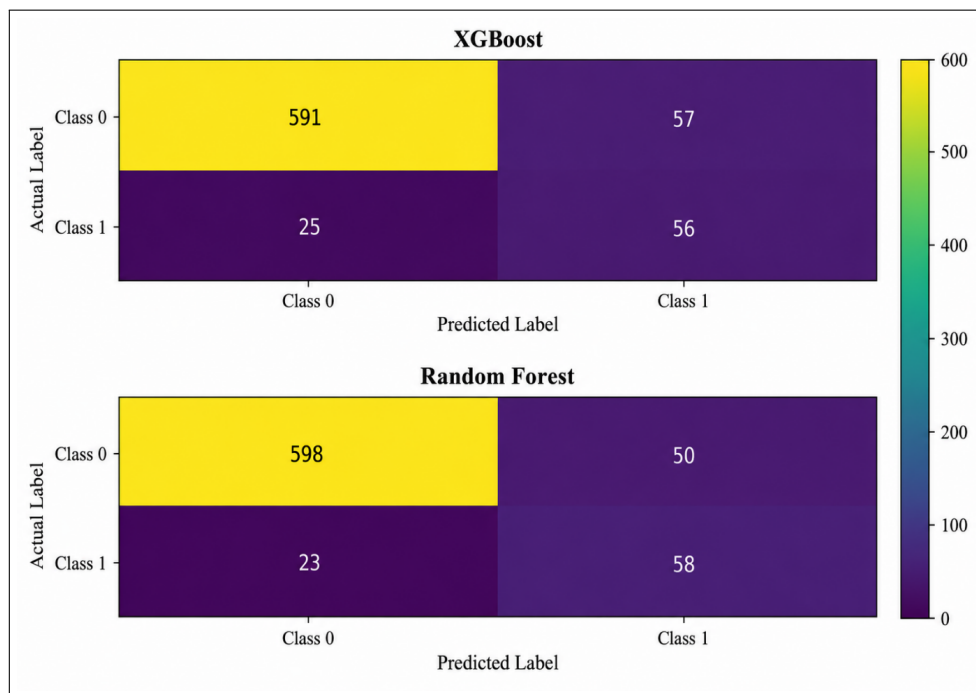


Figure 2. Confusion matrices of the evaluated classification models

Predictive behavior at class level of Random Forest (RF) and XGBoost (XGB) classifiers is examined using the confusion matrices as depicted in Figure 2. Diagonal elements in the matrices are correct predictions, while off-diagonal elements show the number of mispredictions. Misclassification of Non-Dyslexia as Dyslexia class is termed false positive, while misclassification of Dyslexia class as Non-Dyslexia class is called false negative.

4.3 Flask Application

The trained classification models were integrated into a Flask-based application to demonstrate participant-level dyslexia prediction using the proposed machine-learning framework. As shown in Figure 3, the application provides an interactive interface for entering the predictor variables required by the trained models and submitting the participant profile for classification. The processed input is passed to the trained Random Forest and XGBoost models, and the corresponding classification outputs are displayed through the application interface.

Dyslexia Screening Tool
Machine Learning-Based Classification using Random Forest and XGBoost

Home Prediction About

Participant Information
Enter the required predictor variables and submit to obtain dyslexia classification using the trained machine learning models.

Gender: Male
Age: 10
Grade Level: Grade 4
Reading Difficulty: Yes
Writing Difficulty: No
Attention Span: Medium
Memory Retention: Low
Processing Speed: Medium
Phonological Awareness: Low
Academic Performance: Below Average

Predict

Prediction Result

Predicted Class
Dyslexia

Random Forest
Dyslexia
Confidence: 0.67

XGBoost
Dyslexia
Confidence: 0.65

Input Summary

Gender	: Male
Age	: 10
Grade Level	: Grade 4
Reading Difficulty	: Yes
Writing Difficulty	: No
Attention Span	: Medium
Memory Retention	: Low
Processing Speed	: Medium
Phonological Awareness	: Low
Academic Performance	: Below Average

Figure 3. Flask-based Dyslexia classification interface

The application-level integration of the classification pipeline, which includes the input interface and the predicted model output, is shown in Figure 3. The classification presented

here is an output from the trained models using the feature representation in the development of the models. Therefore, the application provides an example of practical implementation of the classification approach.

5. Conclusion and Future Work

The proposed framework provides a systematic binary dyslexia classification method through machine learning based on the individual characteristics acquired from an open-screening test. Both Random Forest and XGBoost algorithms were built with respect to one pre-processing, partitioning, and optimization of hyperparameters procedure to facilitate an analysis of their performance under controlled conditions. Classification accuracy using several metrics along with confusion matrix analysis provides additional insight about performance as well as prediction errors per class. Using Flask to implement the classifiers also provides an illustration of the use of the classifiers in a prediction environment. Nonetheless, performance using a single dataset cannot be generalized to other populations. Future work will focus on external validation, larger samples and demographic diversity, longitudinal studies, explainable machine learning, robust features, and prospective evaluation. Collaboration with validated tests may provide additional analysis on the usefulness of this methodology in both educational and clinical screening environments.

References

- [1] Poornappriya, T. S., and R. Gopinath. "Application of Machine Learning Techniques for Improving Learning Disabilities." *International Journal of Electrical Engineering and Technology (IJEET)*, 2020, vol. 11, no. 10: 392-402.
- [2] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A Scalable Tree Boosting System." In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, 785-794.
- [3] Seshadri, NP Guhan, Sneha Agrawal, Bikesk Kumar Singh, B. Geethanjali, V. Mahesh, and Ram Bilas Pachori. "EEG based Classification of Children with Learning Disabilities Using Shallow and Deep Neural Network." *Biomedical Signal Processing and Control*, 2023, vol. 82: 104553.
- [4] JothiPrabha, A., R. Bhargavi, and BV Deepa Rani. "Prediction of Dyslexia Severity Levels

- from Fixation and Saccadic Eye Movement Using Machine Learning.” *Biomedical Signal Processing and Control*, 2023, vol. 79: 104094.
- [5] Nerušil, Boris, Jaroslav Polec, Juraj Škunda, and Juraj Kačur. ”Eye Tracking-Based Dyslexia Detection Using a Holistic Approach.” *Scientific Reports*, 2021, vol. 11, no. 1: 15687.
- [6] Sbiaa, Fatma, Sonia Kotel, Rania Mghirbi, and Ahmed Ghazi Blaeich. ”Revolutionizing Dyslexia Diagnosis: An Intelligent Model Featuring Machine Learning and Fuzzyfication.” *Procedia Computer Science*, 2024, vol. 246: 3624-3633.
- [7] Liu, Fulin, Hongan Wang, Xin Zhao, and Dongchuan Yu. ”Small-world Properties of Eye-Movement Time Series Assisted in Identifying Children at High Risk for Dyslexia.” *Biomedical Signal Processing and Control*, 2024, vol. 93: 106148.
- [8] Nagarajan, Harshitha, Vishnu Sai Inakollu, Punitha Vancha, and J. Amudha. ”Detection of Reading Impairment from Eye-Gaze Behaviour Using Reinforcement Learning.” *Procedia Computer Science*, 2023, vol. 218: 2734-2743.
- [9] Ding, Ning, Peng Peng, Jiuqing Tang, Yiran Ding, and Jingjing Zhao. ”An Investigation of Phonological Predictors in Chinese Developmental Dyslexia Using a Machine Learning Approach.” *Reading and Writing*, 2025, vol. 38, no. 10: 2849-2873.
- [10] Odoom, Samuel, Eric Opoku Osei, Enock Quansah Effah, Victoria Bofo, Seyram Dusu, and Anthony Kweku Appiah. ”A 3-tier Machine Learning Framework for Early Detection of Learning Difficulties In Basic School Settings.” *Discover Global Society*, 2026, vol. 4, no. 1: 60.
- [11] L. Rello, R. Baeza-Yates, A. Ali, J. P. Bigham, and M. Serra, ”Predicting Risk of Dyslexia with an Online Gamified Test,” *PLOS ONE*, 2020, vol. 15, no. 12: e0241687. Available: <https://www.kaggle.com/datasets/luzrello/dyslexia>