Bulletin of Electrical Engineering and Informatics
Vol. 14, No. 3, June 2025, pp. 2031~ 2041
ISSN: 2302-9285, DOI: 10.11591/eei.v14i3.8639

Ensemble and deep learning via median method for learning disability classification

Anu P. J., K. Ranjith Singh
Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, India

Article Info

Article history:

Received Apr 30, 2024
Revised Nov 6, 2024
Accepted Nov 19, 2024

Keywords:

Deep learning
Ensemble learning model
Learning disabilities
Median combination
Optimizers

ABSTRACT

The study explores the classification of students with and without learning disabilities (LD) through machine learning techniques, utilizing a real dataset and implementing bootstrapping for data augmentation. Noteworthy findings reveal the Adam optimizer's superior performance among various optimizers, achieving a true positive rate (TPR) of 0.97 and a false positive rate (FPR) of 0.02, with high precision, recall, and f1-score values. Additionally, ensemble learning, employing the median method, combines models like Random-ForestClassifier and KerasClassifier, and BaggingClassifier with KerasClassifier, resulting in improved performance. However, the Median-Combined model, integrating AdaBoostClassifier and KerasClassifier, stands out with an accuracy of 99.6%, along with elevated precision, recall, and f1-score values. The comprehensive classification report showcases an overall FPR of 0.0 and TPR of 0.999, highlighting the enhanced performance of the combined model. The significance of this study lies in underscoring the power of fusion between ensemble learning and deep learning techniques, leveraging the median method. This combined model exhibits superior performance, excelling in accuracy, precision, recall, and overall classification effectiveness. The innovative approach of combining both ensemble and deep learning methods through the median method not only advances the understanding of learning disability classification but also emphasizes the practical importance of integrating diverse methodologies for enhanced model performance.

Corresponding Author:

Anu P. J.
Department of Computer Science, Karpagam Academy of Higher Education
Coimbatore, India
Email: anujohnson123@gmail.com

1. INTRODUCTION

Learning disabilities (LD) [1] significantly impact a child's academic and social well-being as a prevalent neurodevelopmental disorder. Early identification and intervention are crucial in enhancing outcomes for children with LD. Despite existing research, there are gaps in understanding the impact of ensemble methods such as bagging or boosting on deep learning models for learning disability detection. Additionally, there is a need to explore the effectiveness of the median fusion method across learning disability contexts and investigate the influence of different deep learning optimizers.

Deep learning techniques hold significant promise for improving diagnosis, intervention, and support strategies for LD. Previous studies, such as those by Dhamal and Mehrotra [2], focused on employing deep learning techniques to predict LD by analyzing relevant data to identify patterns indicative of these disabilities. Ambili and Afsar [3], proposed an artificial neural network (ANN)-based framework for predicting LD using academic performance and cognitive abilities data. Majhi *et al.* [4] presented a comprehensive overview of deep learning applications in neuronal disorders, emphasizing diagnosis, prognosis, and treatment. Kothapalli *et al.* [5] focus on predicting dyslexia and attention deficit hyperactivity disorder (ADHD) using an ensemble classifier model. The research explores the effectiveness of the proposed model in identifying these neurodevelopmental disorders, contributing valuable insights to diagnostic approaches. Sá *et al.* [6] demonstrates significant advancements in the use of machine learning for ASD diagnosis, with impressive results achieved through the application of novel ensemble techniques and gaze anticipation features, offering promising clinical implications for more accurate and accessible diagnosis. The ensemble approach used in this study performs exceptionally well, showing high accuracy and reliability, and underscores the value of using diverse classifiers to improve dyslexia detection [7]. Hang *et al.* [8] addressed the prevalence of diabetes in Malaysian adults, emphasizing the need for accurate prediction tools. Using AdaBoost, support vector machines (SVM), and an ensemble model with feature selection, the research concluded that the ensemble model, combining AdaBoost and SVM, achieved the highest accuracy in predicting diabetes. Mohammed and Kora [9] introduce an innovative ensemble deep learning framework for text classification, addressing the challenge of selecting optimal deep learning classifiers. Their proposed meta-learning ensemble method significantly improves classification accuracy compared to baseline models and outperforms state-of-the-art ensemble methods, demonstrating enhanced performance through the integration of probability distributions for each class label of deep baseline models. The study by Fathi *et al.* [10] addresses the critical need for early Alzheimer's disease (AD) diagnosis using deep learning-based ensemble methods applied to magnetic resonance imaging (MRI) images.

However, these studies often did not address the combination of ensemble learning and deep learning using the median method. There is a lack of exploration into how different ensemble techniques and optimizers can enhance the performance of deep learning models for classifying LD. Specifically, the studies have not comprehensively evaluated the fusion of ensemble and deep learning models to achieve higher classification accuracy and robustness.

The major limitations of existing research include: i) insufficient exploration of the median fusion method in combining ensemble learning and deep learning; ii) limited investigation into the performance of various deep learning optimizers in the context of learning disability detection; and iii) a lack of comprehensive studies on the effectiveness of ensemble methods like bagging and boosting when integrated with deep learning models.

This study aims to fill these gaps by developing and evaluating machine learning models using real data from 348 students. By implementing bootstrapping for data augmentation, testing various optimizers in deep learning models, and combining ensemble learning with deep learning using the median method, this research seeks to improve classification accuracy and reliability. The study provides a novel approach to integrating deep learning and ensemble methods, demonstrating superior performance over individual models.

The significance of this study lies in its innovative approach to combining ensemble learning and deep learning techniques using the median method. This combined model exhibits superior performance, excelling in accuracy, precision, recall, and overall classification effectiveness. It advances the understanding of learning disability classification and emphasizes the practical importance of integrating diverse methodologies for enhanced model performance. The findings provide valuable insights for educators and practitioners to better identify and support students with LD, leading to improved educational outcomes and well-being.

2. METHOD

This study aims to develop a robust predictive model for identifying LD in children using a comprehensive dataset comprising various cognitive and academic attributes. Given the multifaceted nature of LD and the complexity of educational data, our method involves several critical steps: data augmentation, preprocessing, model selection, and performance evaluation.

2.1. Dataset and implementation

In this study, a real-time dataset comprising diverse cognitive and academic attributes of children was utilized. The dataset encompassed features such as age, gender, class, academic performance, and various skills, including reading, writing, spelling, copying, language decoding, fine motor skills, math, attention, hyperactivity, impulsivity, processing speed, auditory discrimination, auditory memory, visual memory, visual discrimination, and visuomotor skills. The dataset consisted of 348 entries with a binary target variable indicating the presence or absence of a LD in children. The model was trained to predict the LD status as either "yes" or "no."

This work is carried out in several stages. Figure 1 shows the stages of the study. First, the dataset is analyzed, and the bootstrapping method is applied to augment data points [11]. Following that, data preprocessing tasks are performed, including dropping irrelevant columns, integer encoding categorical variables, and oversampling to address the class imbalance. The dataset is divided into training and testing sets, with the training set undergoing standardization using a StandardScaler. The mean values obtained are very small and are close to zero, which indicates that the data has been centered around zero. Also, the standard deviation of all 19 features is equal to 1 after the standardization process is done. That means the data has been scaled to a similar range across all the features. Then conducts a comparative analysis of different optimizers in a deep learning model. It involves evaluating multiple optimizers through cross-validation, training the models, making predictions on the test set, and assessing performance metrics including accuracy, receiver operating characteristic-area under the curve (ROC-AUC), and the classification report. The ROC curve plot enables visual comparison of the optimizer's performance, aiding in the identification of the most effective optimizer for the given task.

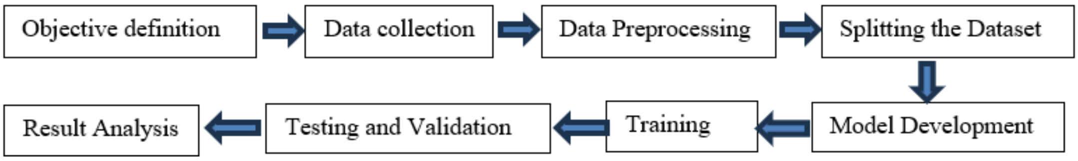


Figure 1. Work process

In the case of a combined model, two individual models such as the ensemble model and the deep learning model are defined. Predictions are made on the test set using both models, and the predictions are combined using the median. Combined model performance is assessed using accuracy, ROC AUC, kappa score, and a classification report, with ROC curves offering a visual performance representation.

2.1.1. Data augmentation

Given the limited data, the bootstrapping method was used to augment [12] the dataset by generating additional data points through resampling. This technique helps to enhance predictive accuracy by increasing the sample size, which mitigates issues associated with insufficient data collection. Specifically, the bootstrapping process involved repeatedly and randomly selecting rows from the original dataset with replacement until the desired count of additional data points was achieved, ensuring that the augmented dataset preserved the original data's distribution.

2.2. Data preprocessing

Data preprocessing involved several steps to prepare the dataset for modeling:

- Dropping irrelevant columns: non-essential columns were removed to streamline the dataset.
- Integer encoding categorical variables: categorical variables were converted to numerical values using integer encoding.
- Oversampling: to address the class imbalance, oversampling techniques were applied to the minority class using the synthetic minority over-sampling technique (SMOTE), which generates synthetic samples by interpolating between existing minority instances.
- Standardization: the dataset was divided into training and testing sets, with the training set undergoing standardization using a StandardScaler. The mean values obtained were very small and close to zero, indicating that the data had been centered around zero. The standard deviation of all 19 features was equal to 1, suggesting that the data had been scaled to a similar range across all features.

2.3. Artificial neural network

ANNs aim to replicate human brain information processing and analysis. They employ brain-inspired models to simulate intricate pattern recognition and prediction tasks. ANNs generally comprise three key elements: an input layer for data input, a hidden layer(s) for computations and transformations, and an output layer for generating the final output [13]. Figure 2 shows the architecture of ANN [14].

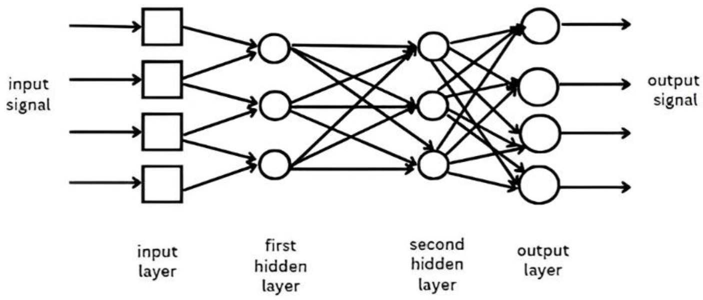


Figure 2. Architecture of ANN

2.3.1. The architecture of an artificial neural network

The ANN used in this study comprises three key layers [15], [16]:

- Input layer: receives input data, with each neuron representing a feature. The size matches the data's dimensionality.
- Hidden layers: these intermediary layers capture intricate data relationships and patterns. The number of hidden layers varies based on the problem's complexity. Neurons in the hidden layers receive inputs from the previous layer, multiply them by weights, and pass the processed information to the next layer. The hidden layer neuron count depends on the problem's complexity and the network's capacity for effective learning and generalization.
- Output layer: provides the final output of the neural network, representing the predictions or classifications made by the network based on the input data and learned patterns.

The study employs the sequential model from Keras, constructing a neural network layer-by-layer [17]. The initial dense layer has 16 neurons with the rectified linear unit (ReLU) activation function. Subsequent layers include dense layers with 8 and 4 neurons using ReLU activation to capture complex data patterns. The final dense layer with 1 neuron employs the sigmoid activation function for binary classification. The model is compiled with popular optimizers (Adam, RMSprop, Adadelta, Adamax, and Adagrad) with specific learning rates contributing to deep learning by updating weights and biases during training. The 'binary_crossentropy' loss function measures dissimilarity between predicted and true labels, and various metrics assess model performance.

The choice of the Adam optimizer is justified by its adaptive learning rate capabilities and its proven performance in various deep learning applications. The ReLU activation function is selected for its ability to mitigate the vanishing gradient problem, allowing the model to learn more effectively. The sigmoid function is appropriate for binary classification tasks, providing probabilities that can be interpreted as class membership.

2.4. Ensemble modeling

Ensemble modeling combines multiple machine learning models to enhance predictive performance by leveraging the diversity and collective intelligence of multiple models for more accurate predictions [18]. The types of ensemble methods used in the study include [19]:

2.4.1. Bagging ensemble method

Bagging trains multiple models independently on varied subsets of data using bootstrap sampling and aggregates their predictions for the final output [20], [21]. This algorithm starts by initializing the number of base estimators and the training dataset. For each base estimator, a bootstrap sample is created by randomly selecting instances with replacements from the training dataset, and the estimator is trained on this sample. During prediction, individual predictions are obtained from each base estimator and aggregated, and the final prediction is based on the combined results.

2.4.2. Boosting ensemble method

Boosting trains models sequentially, with each model correcting the mistakes of the previous ones. The final prediction combines the weighted predictions of all models [21], [22]. This algorithm starts by initializing the training dataset and the number of base estimators. Equal weights are assigned to each training instance. For each base estimator, it is trained on the current weighted training dataset, and the weighted error is calculated. The base estimator's weight is determined using a formula, and instance weights are updated based on correct or incorrect classifications. During prediction, individual predictions from each base estimator are obtained and weighted, and the final prediction is derived by aggregating these weighted predictions.

2.4.3. Random forest ensemble method

Random forest, a popular ensemble method, combines bagging and decision trees. It forms an ensemble of decision trees, with each tree trained on a random subset of features [21], [23]. This algorithm initializes with a training dataset, the number of base estimators, and the maximum number of features to consider at each split. For each base estimator, it creates a random subset of the training dataset through bootstrap sampling, randomly selects a subset of features, and trains a decision tree. These trees collectively form the random forest. During prediction, individual predictions from each decision tree are aggregated to obtain the final prediction. The ensemble methods are chosen for their ability to reduce overfitting and improve generalization by combining the predictions of multiple models. Bagging reduces variance, boosting reduces bias, and random forest combines the strengths of both bagging and decision trees, making them suitable for complex classification tasks.

2.5. Combined model

The study focuses on combining deep learning and ensemble models with median aggregation to improve predictive performance. This involves using a combination of deep learning techniques and ensemble methods to enhance accuracy and robustness.

2.5.1. Implementation of the combined model

This paper focuses on leveraging machine learning techniques to solve a classification problem by combining the strengths of both deep learning and ensemble learning models. The workflow is structured into three main phases:

- Design and train deep learning model: create the model architecture, compile it with a suitable optimizer and loss function, and train it on labeled training data.
- Train ensemble model: choose an ensemble algorithm (random forest, AdaBoost, and bagging), configure it with desired parameters, and train it on the same labeled training data used for the deep learning model.
- Predictions: make predictions on test data using both the trained deep learning model and the ensemble model. Stack the predicted probabilities from both models into a matrix and calculate the median value along the appropriate axis to obtain the combined predictions. This median approach helps capture the central tendency of the predictions from both models, enhancing reliability.

2.5.2. Algorithm for combined model

The combined model leverages the strengths of both deep learning and ensemble methods, capturing complex patterns through deep learning and improving robustness through ensemble techniques. Median aggregation reduces the impact of outliers, leading to more reliable predictions

a. Design and train deep learning model:
   - Define the architecture with appropriate layers and activation functions.
   - Compile the model with an optimizer and loss function.
   - Train the model using the training data.

b. Train ensemble model:
   - Select and configure the ensemble algorithm.
   - Train the ensemble model on the same training data.

c. Make predictions:
   - Use both models to make predictions on the test data.
   - Stack the predicted probabilities from both models.
   - Calculate the median of the stacked probabilities to obtain combined predictions.

d. Evaluate performance:
   - Calculate evaluation metrics such as accuracy, ROC AUC, kappa score, and generate a classification report.
   - Plot ROC curves to visually compare model performance.

3. RESULTS AND DISCUSSION

This section introduces the performance evaluation of the proposed architecture for learning disability classification.

3.1. Evaluation matrices

To measure the system's performance, various parameters are used, such as precision, recall, f1-score, [24] kappa measurement, ROC-AUCscore, true positive rates (TPR), false positive rate (FPR), and accuracy [25], [26].

a. Precision

$$Precision = true\ positive/(true\ positive + false\ positive) \quad (1)$$

b. Recall:

$$Recall = true\ positive/(true\ positive + false\ negative) \quad (2)$$

c. F1-score:

$$F1 - score = 2 * (precision * recall)/(precision + recall) \quad (3)$$

d. Accuracy:

$$Accuracy = (true\ positive + true\ negative)/(true\ positive + true\ negative + false\ poistive + false\ negative) \quad (4)$$

e. TPR

$$TPR = true\ positive/(true\ positive + false\ negative) \quad (5)$$

f. FPR

$$FPR = false\ negative/(false\ positive + true\ negative) \quad (6)$$

g. Kappa measurement

It assesses agreement between observed and expected classifications, accounting for the possibility of chance agreements, with higher coefficients indicating stronger agreement or better performance.

h. ROC-AUC score

It evaluates binary classifier performance by plotting TPR against FPR at various thresholds, with higher values representing superior performance.

3.2. Result analysis

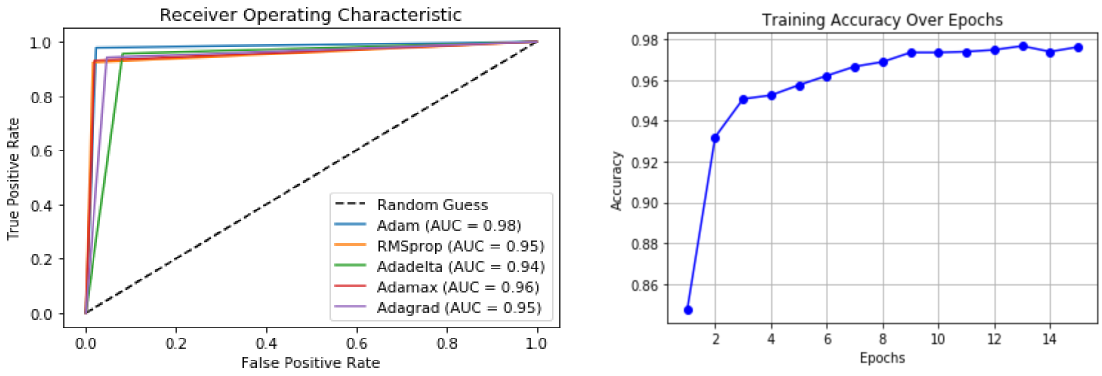
3.2.1. Performance evaluation of artificial neural network models with different optimizers

The performance of the ANN models with various optimizers was evaluated using several metrics, including precision, recall, f1-score, Kappa value, ROC-AUC, TPR, FPR, and accuracy. The results are summarized in Table 1. The ANN-Adam model outperformed others across multiple performance metrics. It showcased exceptional precision, recall, and f1-score of 98%, indicating its superior ability to correctly classify positive instances. The high Kappa value of 95.49 reflects the robustness of ANN-Adam, and its impressive ROC-AUC of 97.74 indicates its excellent ability to distinguish between positive and negative instances. The model achieved a remarkable accuracy of 97.82%, with a very low FPR of 0.02%. These findings highlight the effectiveness of the Adam optimizer in training deep learning models for classification tasks, especially in the context of LD.

Table 1. Performance evaluation metrics of the ANN with different optimizers

Algorithm	Precision	Recall	F1-score	Kappa value	ROC-AUC	TPR	FPR	Accuracy
ANN-Adam	98	98	98	95.49	97.74	97.82	0.02	97.74
ANN-RMSprop	96	95	95	90.62	95.41	92.39	0.01	95.3
ANN-Adadelta	94	94	94	87.58	93.47	95.65	0.08	93.8
ANN-Adamax	96	95	95	90.99	95.58	93.11	0.01	95.49
ANN-Adagrad	95	95	95	89.48	94.76	94.2	0.04	94.74

Figure 3 shows ROC-AUC of different optimizers and epochs for Adam optimizer. Figure 3(a) specially illustrates the ROC-AUC scores for ANN models using various optimizers. This visualization highlights the performance comparison between different optimization algorithms, providing insight into their efficacy in model training. On the other hand, Figure 3(b) captures the training dynamics of the ANN model specifically employing the Adam optimizer. It plots the accuracy against epochs, showcasing the model's learning progression over time. This subplot emphasizes how the Adam optimizer influences the model's convergence and accuracy, offering a detailed view of its impact on the training process.



(a) (b)

Figure 3. ROC-AUC of different optimizers and epochs for Adam optimizer; (a) the ROC-AUC scores plotted for the ANN models utilizing different optimizers and (b) the training process over epochs and accuracy, plotted for the Adam optimizer

3.2.2. Cross-validation scores for artificial neural network models with different optimizers

Cross-validation was used to assess the consistency of the ANN models. The cross-validation scores for different optimizers are shown in Table 2. Cross-validation is a technique used to assess the performance of a machine learning model by dividing the dataset into multiple subsets, training the model on a portion of the data, and evaluating it on the remaining portion [27].

Table 2. The cross-validation scores for ANN models with different optimizers

Algorithm	Mean
ANN-Adam	0.961
ANN-RMSprop	0.940
ANN-Adadelta	0.954
ANN-Adamax	0.931
ANN-Adagrad	0.900

When comparing models based on mean and standard deviation, a lower standard deviation indicates less variability in the performance across different subsets, suggesting higher consistency. A higher mean score indicates better average performance. Therefore, a model with a higher mean score and a lower standard deviation is generally considered to have performed well and to be more reliable.

The ANN-Adam model demonstrated the highest mean score, indicating its robustness and consistency across different data subsets. ANN-RMSprop and ANN-Adadelta followed closely in mean performance, while ANN-Adagrad, though consistent, had a lower mean score. This suggests that while all optimizers provided good performance, Adam consistently outperformed others in terms of both stability and accuracy.

3.2.3. Performance evaluation of combined models

The combined models' performance was evaluated using various classifiers, as summarized in Table 3. Table 3 shows the analysis of three different classifiers, random forest with KerasClassifier, AdaBoost with KerasClassifier, and bagging with KerasClassifier, it is observed that all three models displayed remarkable performance across various evaluation metrics. Notably, the AdaBoost classifier delivered perfect precision, recall, and f1-score, signifying an exceptional ability to correctly classify positive and negative cases. In comparison, the random forest and bagging classifiers also exhibited high precision, recall, and f1-score, with minimal differences in performance. The Kappa values indicated strong agreement between the predicted and actual values for all three models. Furthermore, the classifiers showcased exceptional discrimination power, as reflected by their high ROC-AUC values, with AdaBoost leading in this regard. The TPR were consistently high for all three, while the FPR were impressively low, signifying a minimal misclassification of negative cases as positive. The accuracy was notably high for all three classifiers.

The AdaBoost+KerasClassifier model stood out with perfect precision, recall, and f1-score, indicating its exceptional ability to classify both positive and negative cases correctly. The high Kappa value of 99.24 and ROC-AUC of 99.98 further demonstrate its superior performance. The model achieved an accuracy of 99.62%, highlighting its robustness and reliability. These results indicate that the combination of

AdaBoost and KerasClassifier is highly effective for classification tasks, leveraging the strengths of both ensemble learning and deep learning.

Table 3. Performance evaluation of combined model

Algorithm	Precision	Recall	F1-score	Kappa value	ROC-AUC	TPR	FPR	Accuracy
RandomForest+KerasClassifier	99	99	99	98.12	100	98.18	0.0	99.06
AdaBoost+KerasClassifier	100	100	100	99.24	99.98	99.27	0.0	99.62
Bagging+KerasClassifier	99	99	99	97.21	99.97	97.18	0.0	98.60

Figure 4 presents the ROC-AUC score plots for different ensemble models combined with KerasClassifier. Figure 4(a) demonstrates the performance of the random forest combined with KerasClassifier, indicating how well this ensemble method can classify the data. Figure 4(b) shows the ROC-AUC scores for the bagging approach paired with KerasClassifier, illustrating the impact of this ensemble technique on classification performance. Lastly, Figure 4(c) displays the ROC-AUC scores for the Adaboost combined with KerasClassifier, providing insights into the effectiveness of this boosting method in enhancing the model's predictive accuracy. Each subplot offers a comparative view of the ROC-AUC scores, facilitating an understanding of how different ensemble strategies perform in conjunction with the KerasClassifier.

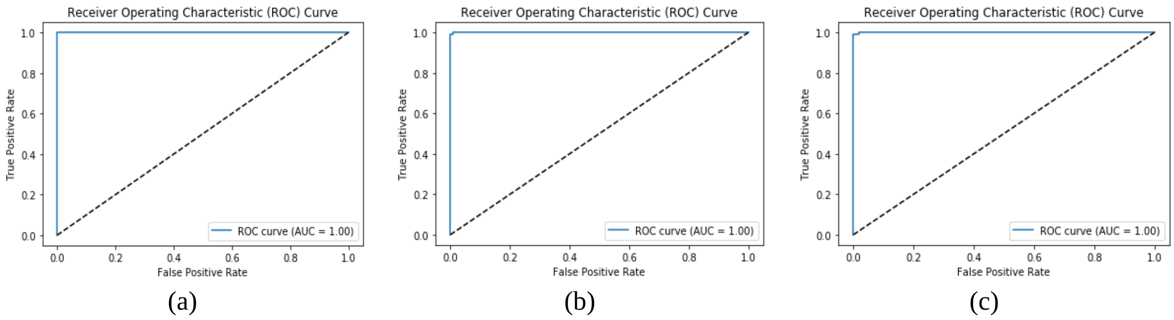


Figure 4. ROC-AUC score plotting for the; (a) RandomForest+KerasClassifier, (b) Bagging+KerasClassifier, and (c) Adaboost+KerasClasifier models

3.3. Comparison with other studies and interpretation of findings

The findings of this study align with previous research that highlights the effectiveness of combining ensemble methods with deep learning models. Studies such as Sá *et al.* [6] and Zaree *et al.* [7] have demonstrated the potential of ensemble learning in improving classification performance. The superior performance of the combined AdaBoost+KerasClassifier model in this study further validates the effectiveness of such hybrid approaches. For instance, Sá *et al.* [6] demonstrated that ensemble methods could significantly enhance model accuracy, while Zaree *et al.* [7].

Zaree *et al.* [7] emphasized the robustness of combined models in handling diverse datasets. The results demonstrate that combining deep learning with ensemble methods using median aggregation can significantly enhance the classification accuracy and robustness of models for learning disability detection. The AdaBoost+KerasClassifier model's perfect precision, recall, and f1-score suggest that this approach effectively mitigates false positives and negatives, providing reliable predictions. This approach not only enhances accuracy but also provides robustness against variations in the dataset, making it a versatile method for various classification tasks.

3.4. Study limitations and future directions

Despite the promising results, this study has some limitations. The dataset size is relatively small, which may affect the generalizability of the findings. Future research should consider using larger and more diverse datasets to validate the effectiveness of the proposed models. Additionally, the study primarily focused on binary classification; exploring multi-class classification could provide further insights into the model's capabilities. Future research can explore the use of other ensemble techniques, such as gradient boosting, voting, or stacking, to further enhance model performance. Investigating the application of these combined models in different domains and problem scenarios could validate their effectiveness and extend their applicability. Incorporating feature selection methods and experimenting with different data augmentation techniques could also provide valuable insights. Research can also delve into real-time applications of these models, particularly in educational settings, to assess their practical utility in early detection and intervention of LD.

4. CONCLUSION

This study underscores the significant potential of integrating deep learning with ensemble methods to enhance the accuracy and robustness of models in detecting LD. By utilizing the Adam optimizer in ANN models and combining AdaBoost with KerasClassifier, the research achieved superior performance with high precision, recall, and f1-scores, alongside exceptional accuracy and robustness. These results highlight the effectiveness of leveraging diverse methodologies like ensemble learning and deep learning to significantly improve classification tasks, facilitated by the median aggregation method for synthesizing model strengths. The enhanced classification accuracy of these combined models suggests promising strides in early detection of LD, potentially leading to timely interventions and better educational outcomes. These models can serve as reliable screening tools for educators and practitioners, applicable not only in educational settings but also in domains requiring precise classification such as medical diagnoses and fraud detection. The broad applicability of these methodologies underscores their versatility and robustness across different applications.

Moving forward, future research should concentrate on validating these findings with larger, more diverse datasets to ensure the models' generalizability across various populations and settings. Exploring multi-class classification applications could further elucidate the capabilities and limitations of the proposed models. Investigating various feature selection techniques and optimizing resource utilization through advanced algorithms and hyperparameter tuning are also crucial for enhancing model performance and efficiency in real-world scenarios.

Integrating ensemble learning with deep learning models significantly boosts classification performance, confirming the hypothesis that a combined approach leverages the strengths of individual models while mitigating their weaknesses. However, the study also highlights the challenges of model complexity and computational demands, suggesting a need for optimizing resource utilization. Comparing these findings with previous research shows that while traditional models have their merits, the combination of methods as proposed in this study offers a distinct advantage in terms of accuracy and reliability. This positions the combined model approach as a promising direction for future research and application in various fields.

ACKNOWLEDGEMENTS

Special thanks are extended to Karpagam Academy of Higher Education, Coimbatore, India for giving guidelines for this research.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Anu P. J.	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
K. Ranjith Singh		✓			✓		✓	✓		✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

ETHICAL APPROVAL

Not applicable.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [Anu P. J.], upon reasonable request. The data, which contain information that could compromise the privacy of research participants, are not publicly available due to certain restrictions.

REFERENCES

[1] T. Aro, R. Neittaanmäki, M. Torppa, E. Korhonen, H. Riihimäki, and M. Torppa, "A Register Study Suggesting Homotypic and Heterotypic Comorbidity Among Individuals with Learning Disabilities," *Journal of Learning Disabilities*, vol. 57, no. 1, pp. 30-42, 2023, doi: 10.1177/00222194221150230.
[2] P. Dhamal and S. Mehrotra, "Deep Learning Approach for Prediction of Learning Disability," in *2021 5th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence*, 2021, pp. 77-83, doi: 10.1145/3461598.3461611.
[3] K. Ambili and P. Afsar, "A Framework for Learning Disability Prediction in School Children Using Artificial Neural Network," *International Journal of Advanced Research in Science, Engineering and Technology*, vol. 3, no. 6, pp. 2312- 2316, 2016.
[4] V. Majhi, A. Saikia, A. Datta, A. Sinha, and S. Paul, "Comprehensive Review on Deep Learning for Neuronal Disorders: Applications of Deep Learning," *International Journal of Natural Computing Research (IJNCR)*, vol. 9, no. 1, p. 18, 2020, doi: 10.4018/IJNCR.2020010103.
[5] P. K. V. Kothapalli, V. Rathikarani, and G. K. M. Nookala, "Prediction of Dyslexia and Attention Deficit and Hyperactivity Disorder Using Ensemble Classifier Model," *International Journal of Systems Assurance Engineering and Management*, 2022, doi: 10.1007/s13198-022-01724-z.
[6] R. O. da S. Sá *et al.*, "Enhancing Ensemble Classifiers Utilizing Gaze Tracking Data for Autism Spectrum Disorder Diagnosis," *Computers in Biology and Medicine*, vol. 182, 2024, p. 109184, doi: 10.1016/j.compbiomed.2024.109184.
[7] M. Zaree, M. Mohebbi, and R. Rostami, "An Ensemble-Based Machine Learning Technique for Dyslexia Detection During a Visual Continuous Performance Task," *Biomedical Signal Processing and Control*, vol. 86, Part B, 2023, p. 105224, doi: 10.1016/j.bspc.2023.105224.
[8] O. Y. Hang, V. Wiwied, and R. Rosaida, "Diabetes Prediction Using Machine Learning Ensemble Model," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 37, no. 1, pp. 82-98, 2024, doi: 10.37934/araset.37.1.8298.
[9] A. Mohammed and R. Kora, "An Effective Ensemble Deep Learning Framework for Text Classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 8825-8837, 2022, doi: 10.1016/j.jksuci.2021.11.001.
[10] S. Fathi, A. Ahmadi, A. Dehnad, M. Almasi-Dooghaee, and M. Sadegh, "A Deep Learning-Based Ensemble Method for Early Diagnosis of Alzheimer's Disease Using MRI Images," *Neuroinformatics*, vol. 22, pp. 89-105, 2024, doi: 10.1007/s12021-023-09646-2.
[11] M. Nakhwan and R. Duangsoithong, "Comparison Analysis of Data Augmentation using Bootstrap, GANs and Autoencoder," *2022 14th International Conference on Knowledge and Smart Technology (KST)*, Chon buri, Thailand, 2022, pp. 18-23, doi: 10.1109/KST53302.2022.9729065.
[12] A. Kebaili, J. Lapuyade-Lahorgue, and S Ruan, "Deep Learning Approaches for Data Augmentation in Medical Imaging: A Review," *Journal of Imaging*, vol. 9, no. 4, p. 81, 2023, doi: 10.3390/jimaging9040081.
[13] A. Abraham, "Artificial Neural Networks," in *Handbook of Measuring System Design*, P. H. Sydenham, Ed. Chichester, U.K.: John Wiley & Sons, 2005.
[14] T. Das, "Machine Learning Algorithms for Image Classification of Hand Digits and Face Recognition Dataset," *International Research Journal of Engineering and Technology*, vol. 4, no. 12, pp. 640-649, Dec. 2017.
[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436–444, 2015, doi: 10.1038/nature14539.
[16] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Neural Networks*, vol. 61, pp. 85-117, 2015, doi: 10.1016/j.neunet.2014.09.003.
[17] R. Kasemi and B. Karahoda, "Implementation of Artificial Neural Network in Embedded Systems," in *UBT International Conference 2020*, 2020
[18] D. J. Lieske, M. S. Schmid, and M. Mahoney, "Ensembles of ensembles: combining the predictions from multiple machine learning methods," *Machine Learning for Ecology and Sustainable Natural Resource Management*, pp. 109-121, 2018, doi: 10.1007/978-3-319-96978-7_5.
[19] J. Obregon and J. Y. Jung, "Explanation of Ensemble Models," in *Human-Centered Artificial Intelligence*, pp. 51-72, 2022, doi: 10.1016/B978-0-323-85648-5.00011-6.
[20] G. Ngo, R. Beard, and R. Chandra, "Evolutionary Bagging for Ensemble Learning," *Neurocomputing*, vol. 510, pp. 1-14, 2022, doi: 10.1016/j.neucom.2022.08.055.
[21] I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects," in *IEEE Access*, vol. 10, pp. 99129-99149, 2022, doi: 10.1109/ACCESS.2022.3207287.
[22] R. E. Schapire, "The Boosting Approach to Machine Learning: An Overview," in *Nonlinear Estimation and Classification*, vol. 171, pp. 149-171, 2003, doi: 10.1007/978-0-387-21579-2_9.

[23] Y. Mishina, M. Tsuchiya, and H. Fujiyoshi, "Boosted Random Forest," in *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, Lisbon, Portugal, pp. 594-598, 2014, doi: 0.5220/0004739005940598.

[24] Y. Alraba'nah and W. Toghuj, "A Deep Learning-Based Architecture for Malaria Parasite Detection," *Bulletin of Electrical Engineering and Informatics*, vol. 13, pp. 292-299, 2024, doi: 10.11591/eei.v13i1.5485.

[25] J. Cohen, "A Coefficient of Agreement for Nominal Scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37-46, 1960, 10.1177/001316446002000104

[26] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves," in *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pp. 233-240, 2006.

[27] I. Tougui, A. Jilbab, and J. El Mhamdi, "Impact of the Choice of Cross-Validation Techniques on the Results of Machine Learning-Based Diagnostic Applications," *Healthcare Informatics Research*, vol. 27, no. 3, pp. 189-199, 2021, doi: 10.4258/hir.2021.27.3.189.

BIOGRAPHIES OF AUTHORS

Anu P. J. received his B.Sc. degree in computer science from Calicut University, Kerala, India in 2010. In 2013, she completed MCA. She graduated in June 2013 with an excellent percentage. She worked as a lecturer in the Faculty of Computer Science at DonBosco College, affiliated with Calicut University for 7 years. Her main research interests include artificial intelligence. She can be contacted at email: anujohnson123@gmail.com.

K. Ranjith Singh obtained his Ph.D. degree from Periyar University, Salem, Tamil Nadu, India. He is working as Head of the Department, Department of Computer Technology, Faculty of Arts, Science, Commerce and Management, Karpagam Academy of Higher Education (Deemed to be University), Coimbatore, Tamil Nadu, India. He has more than 21 years of experience in the academic field. He has published several research articles in reputed International Journals (Scopus) and International Conferences (Scopus) so far. His research areas of interest include cryptography and network security, mobile ad-hoc networks, and wireless sensor networks. He can be contacted at email: ranjithsingh.koppaiyan@kahedu.edu.in.