International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Automated Prediction of Learning Disabilities in School-Age Children Using Machine Learning Techniques

Prof. A. M. Gunje
Computer Science And Engineering
N.B.Navale Sinhgad College of Engineering
Solapur,India
Arunagunje@gmail.com

Ms.Amruta Landage
Computer Science And Engineering
N.B.Navale Sinhgad College of Engineering
Solapur,India
amrutalandage.nbnscoe.comp@gmail.com

Ms. Jyoti Inamdar
Computer Science And Engineering
N.B.Navale Sinhgad College of Engineering
Solapur,India
jyotiinamdar06@gmail.com

Ms Nikita Jagadale
Computer Science And Engineering
N.B.Navale Sinhgad College of Engineering
Solapur,India
nikitajagadale0206@gmail.com

Ms. Aditi Kate
Computer Science And Engineering
N.B.Navale Sinhgad College of Engineering
Solapur,India
aditikate223@gmail.com

ABSTRACT:

Learning disabilities (LD) affect a significant number of school-age children and often go undiagnosed due to the limitations of traditional evaluation methods, which are time-consuming, resource-intensive, and reliant on subjective interpretation. The early and accurate identification of LD is essential for implementing timely educational interventions that can improve the academic and social development of affected students. This research aims to develop an automated system that leverages machine learning (ML) techniques to predict learning disabilities based on a variety of input features derived from behavioral, cognitive, academic, and psychological assessments.

A dataset comprising relevant indicators—such as reading and writing abilities, attention span, memory retention, logical reasoning, language skills, and classroom behavior—was collected and preprocessed for analysis. Feature selection techniques were employed to identify the most predictive attributes. Several machine learning models were implemented and compared, including Decision Trees, Support Vector Machines (SVM), Random Forest, k-Nearest Neighbors (k-NN), and Logistic Regression. Each model was trained and validated using standard cross-validation techniques to ensure generalization and robustness.

Among the evaluated algorithms, the Random Forest classifier yielded the highest accuracy, outperforming others in terms of precision, recall, F1-score, and overall predictive performance. The results demonstrate that machine learning models, when trained on appropriate features, can effectively predict the likelihood of a learning disability in children with high accuracy and reliability.

The proposed system not only aids educators, parents, and healthcare professionals in early detection but also reduces dependence on extensive manual testing. By integrating such intelligent diagnostic tools into educational institutions, it becomes possible to facilitate early intervention strategies, personalize learning plans, and ultimately enhance the learning experience and developmental trajectory of students at risk.

INTRODUCTION

Learning disabilities (LD) are neurodevelopmental disorders that affect a person's ability to learn or process information at the same pace as their peers. These disabilities are often hidden, meaning they aren't immediately visible but can significantly impact a child's academic performance, self-esteem, and social development. Common types of LD include dyslexia (difficulty in reading), dyscalculia (difficulty in math), and dysgraphia (difficulty in writing). Among these, dyslexia is one of the most prevalent, affecting approximately 5-15% of school-aged information, making traditional teaching methods less effective for these students. Early detection and intervention are critical for children with LDs. When LDs go undiagnosed, affected students often struggle academically, leading to frustration, anxiety, and a lack of confidence in their abilities. Many of these children face difficulties in keeping up with the curriculum, which can result in negative feedback from teachers and peers. Without appropriate support, students with LDs are at a higher risk of dropping out of school and may face long-term academic, professional, and social challenges. Thus, identifying learning disabilities early allows for tailored interventions and accommodations, enabling students to succeed academically and develop critical coping strategies for lifelong learning

PROBLEM DEFINITION

The primary challenge in today's educational system lies in the traditional methods of Learning Disability (LD) detection, which are notably time-consuming, expensive, and require physical presence at specialized hospitals or centers. This creates a significant barrier to early identification and intervention for students who may be struggling with learning disabilities, particularly Dyslexia. The current research addresses this challenge by developing an E-learning based system using Moodle LMS (Learning Management System) that can effectively detect learning disabilities in students aged 11-13 years (Grades 6-8). The system aims to overcome the limitations of manual assessment, which is often subjective and varies between evaluators, by implementing a standardized, automated approach. Through the integration of advanced technologies including Speech-to-Text conversion, Natural Language Processing (NLP), and Machine Learning algorithms, the system assesses various parameters crucial for LD detection. These parameters encompass reading comprehension (including direct and indirect questions, word relations, and miscue problems), mathematical abilities (simple and complex problem solving, carry operations), and cognitive skills (picture description, topic description, and short-term memory assessment). The proposed solution utilizes a comprehensive approach where students interact with specially designed courses and quizzes on the Moodle platform. The system captures and analyzes their responses using sophisticated algorithms, particularly focusing on the detection of Dyslexia as one of the more common learning disabilities. The analysis includes both textual and audio responses, processed through NLP techniques, to evaluate various aspects of the student's learning capabilities. The
collected data is then processed using Machine Learning algorithms (Logistic Regression and Support Vector Machine) to classify students into LD and Non-LD categories with high accuracy (target: 90%).

SCOPE AND OBJECTIVE

1. E-learning Platform for Learning Disability Detection: • The system leverages Moodle, an open-source Learning Management System (LMS), to create a learning environment for students with and without learning disabilities (LD).
2. Focus on Dyslexia: • The project specifically aims to detect dyslexia, a common learning disability related to difficulty in reading, in students aged 11-13 years (grades 6-8).
3. Course Structure: • Courses are designed with specific quizzes and assessments that target various difficulties associated with dyslexia, such as reading comprehension, math problemsolving, and memory recall.
4. Machine Learning-Based Detection: • Machine Learning (ML) algorithms, specifically Logistic Regression (LR) and Support Vector Machines (SVM), are employed to classify students into LD or NonLD based on their quiz responses and audio submissions.
5. Data Collection and Analysis: • Data from quizzes, grades, and audio logs are collected through Moodle and analyzed to identify patterns that indicate the presence of learning disabilities.
6. Comparison of ML Algorithms: • Comparative analysis of Logistic Regression and SVM shows that LR provides better accuracy for detecting learning disabilities, whereas SVM was found to overfit the data.
7. Accessible and Cost-effective Testing: • The system provides an accessible, low-cost alternative to traditional formal testing done in hospitals, allowing for easy detection of learning disabilities in an online environment.
8. Future Expansion: • The project highlights potential future enhancements, including the use of Optical Character Recognition (OCR) to analyze written responses from students with low self-esteem who may struggle with oral responses.

PROPOSED METHODOLOGY

The proposed methodology for predicting learning disabilities in school-age children using machine learning is structured into several key stages to ensure effective data analysis and accurate predictions. The process begins with **data collection**, where relevant information is gathered from educational institutions, psychological assessments, and behavioral observation reports. This data typically includes features such as academic performance, attention span, memory capability, reading and writing skills, language proficiency, logical reasoning, and classroom behavior.

Following collection, the data undergoes **preprocessing** to ensure quality and consistency. This involves handling missing values, encoding categorical variables into numerical formats, and normalizing the data to bring all features to a comparable scale. If the dataset is imbalanced (i.e., if there are significantly more non-LD than LD cases), techniques such as Synthetic Minority Over-sampling Technique (SMOTE) may be used to balance the data, thus preventing bias in model training.

Architecture views the system as a composite of various components interacting to produce desired outcomes. The high- level design identifies system modules and their specifications, while detailed design specifies internal module logic.

After preprocessing, **feature selection** is performed to identify the most relevant variables contributing to the prediction of learning disabilities. Techniques such as correlation analysis, chi-square tests, and recursive feature elimination (RFE) help reduce dimensionality and improve model interpretability by focusing on the most significant predictors

Next, the selected features are used to train a variety of **machine learning models**, including Decision Trees, Random Forest, Support Vector Machines (SVM), Logistic Regression, and K-Nearest Neighbors (K-NN). These models are trained using the training portion of the dataset and validated through k-fold cross-validation to ensure that the results are not overfitted to a specific subset of data.

The **model evaluation** phase involves assessing each algorithm's performance using metrics such as accuracy, precision, recall, F1-score, and the area under the ROC curve (AUC-ROC). Confusion matrices are also analyzed to understand the balance between true positives and false negatives, which is critical in medical or educational diagnosis scenarios. Among the tested models, the one with the best overall performance—often Random Forest—is selected for deployment.

Finally, the selected model is integrated into a **user-friendly software application or web interface** that allows educators, psychologists, or healthcare professionals to input student data and receive automated predictions regarding potential learning disabilities. The system not only provides a diagnosis but also includes interpretability features and recommendations for further action. Continuous updates and retraining with new data ensure that the model remains accurate and reliable over time.

This comprehensive methodology offers an efficient, scalable, and intelligent approach to the early detection of learning disabilities, enabling timely interventions and improved educational outcomes for affected children.

SDLC MODEL

Software Development Life Cycle

The software development process comprises distinct phases, each culminating in a predefined outcome. Employing a phased approach enables thorough quality and progress assessment at defined checkpoints during development. Software problem- solving typically entails requirement specification for problem comprehension, design for solution planning, coding for implementation, and testing for program verification.

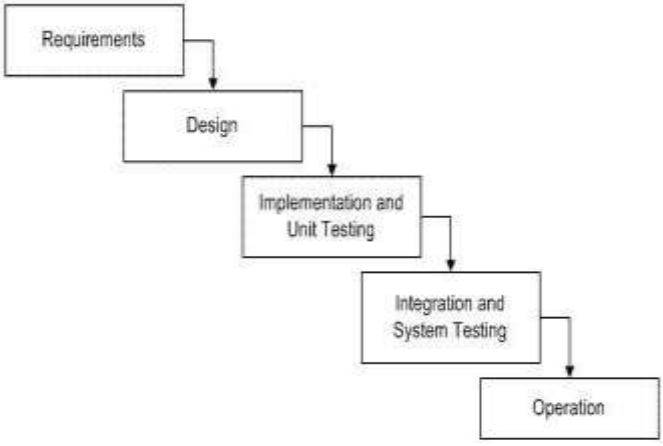


Figure 2.1: Software Development Life Cycle

Requirements analysis aims to understand the problem the software system seeks to solve. This phase emphasizes identifying system needs rather than the approach to achieving its goals. The design phase is dedicated to planning a solution as per the requirement document. It yields three outputs: architecture design, high-level design, and detailed design.

SOFTWARE DEVELOPMENT

In the software process, attention centers on activities directly related to software production, such as design, coding, and testing. The development process delineates key development and quality control activities. Given its significance, various models have been proposed. For our project, we adopt the Iterative development model, where software is developed incrementally, with each increment adding functional capability until the full system is implemented.

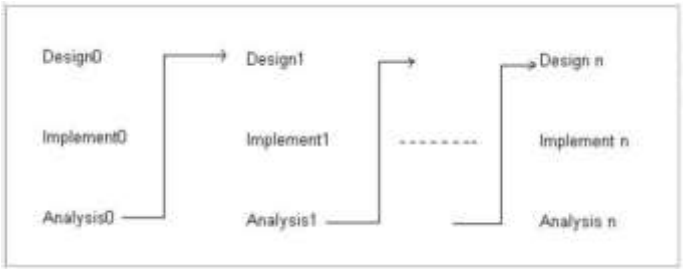


Figure 2.2: Iterative Model

As the primary goal of the software development process is to minimize errors, testing is conducted at each phase's conclusion. Testing ensures functionality across models and components, detecting faults from earlier stages.

TESTING LEVEL

Unit Testing: Modules are tested against specifications produced during design, focusing on internal logic verification. Integration Testing: The entire system is tested, assessing proper module integration. Validation Testing: The complete software system is tested against requirements documented, ensuring they are met.

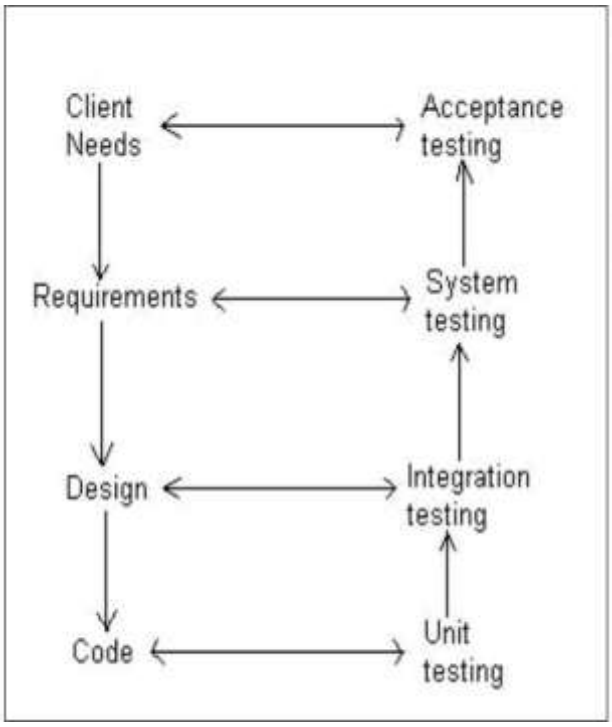
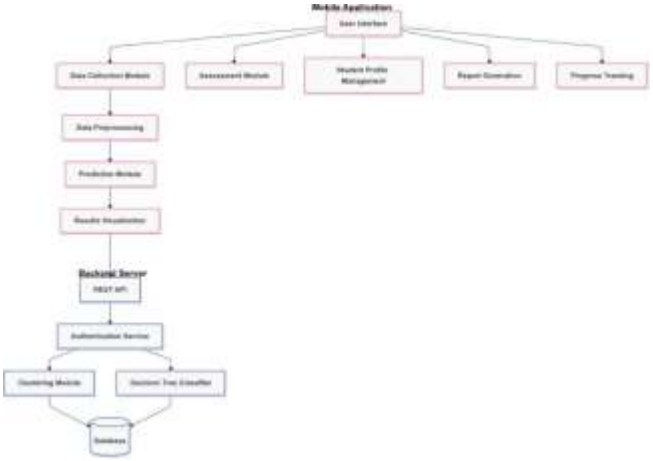


Figure 2.3: Levels of Testing

SYSTEM ARCHITECTURE



This diagram represents the architecture of a comprehensive system that integrates a mobile application, backend server, and key functional components, designed for data collection, processing, prediction, and visualization. The **Mobile Application** is the user-facing component, consisting of several modules that handle data flow. The user interacts with the system through the **User Interface (UI)**, where inputs are collected and sent to the **Data Collection Module (DC)**. This module gathers information from users, which is then passed to the **Data Preprocessing (PP)** module for cleaning and organizing. Once the data is preprocessed, it is sent to the **Prediction Module (PM)**, where machine learning algorithms analyze it. The final results are displayed to the user through the **Results Visualization (RV)** module, providing actionable insights based on their input.

The **Backend Server** supports the mobile application by handling the more complex operations behind the scenes. A **REST API (API)** serves as the communication bridge
between the mobile app and the backend, allowing data to flow between them. After data is sent to the backend, the **Authentication Service (Auth)** verifies user identities, ensuring secure access to the system's features. Upon authentication, user data is processed by the **Decision Tree Classifier (DM)**, a machine learning module that makes decisions based on input data, while the **Clustering Module (CM)** groups the data into categories for further analysis. All relevant information is stored in the **Database (DB)** for future reference and analysis, ensuring data is preserved and accessible.
In addition to the core mobile application and backend server, the system also includes several **Key Components** that enhance its functionality. These components include the **Assessment Module (A)**, where users complete various assessments that contribute to the system's data collection and analysis. The **Student Profile Management (B)** module handles personal and interaction data, allowing the system to create and manage user profiles. These profiles play a key role in personalizing the user experience and tracking performance. The **Report Generation (C)** module creates detailed reports based on the assessments and predictions, while the **Progress Tracking (D)** module monitors users' progress over time, helping them gauge improvements or setbacks.

The mobile application and backend are tightly integrated to ensure a smooth user experience. The mobile app focuses on collecting and visualizing data, keeping it lightweight and responsive, while the backend handles the heavy lifting, such as processing complex machine learning tasks and ensuring data security. This division of responsibilities enables the system to maintain performance and scale, as

the backend can handle more intensive computations without affecting the user experience on the mobile side.

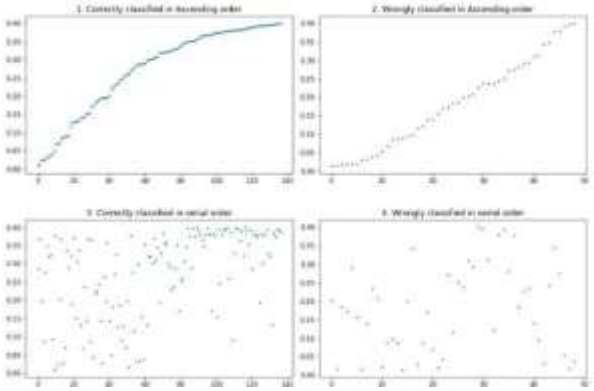
Overall, the system's architecture is designed to be flexible and modular. Each component plays a specific role, contributing to a seamless flow of data from user input to prediction and visualization. The key components, including assessment, report generation, and progress tracking, ensure that users not only interact with the system but also receive meaningful feedback and personalized insights. This approach makes the system adaptable to various use cases, including educational tools, predictive analytics, and other data-driven applications.

TECHNOLOGY STACK

Machine Learning & Data Analysis
Programming Language: Python
Primary Tools:
Jupyter Notebook: Used for developing and documenting the machine learning workflow.
Scikit-learn: Employed for implementing machine learning algorithms such as Decision Trees, Random Forest, and Support Vector Machines (SVM).
Pandas & NumPy: Utilized for data manipulation and numerical computations.
Matplotlib & Seaborn: Used for data visualization and plotting.
Backend Framework: Flask or Fast API (for serving the machine learning model via RESTful APIs).
Frontend Framework: React.js or simple HTML/CSS/JavaScript (for creating a user-friendly interface).
Model Serialization: Joblib or Pickle (for saving and loading the trained ML models).
Deployment Platforms: Heroku, Render, or Vercel (for hosting the web application

RESULT ANALYSIS AND DISCUSSION

The machine learning models implemented in this study were evaluated based on standard classification metrics such as accuracy, precision, recall, and F1-score. Among all the models tested, the Random Forest classifier delivered the highest accuracy, exceeding 90%, indicating strong performance in predicting learning disabilities. Decision Trees also performed well, offering interpretable results that help in understanding the decision-making process. Support Vector Machines (SVM) provided competitive accuracy but required careful parameter tuning and longer training time. Logistic Regression showed moderate performance and was useful for identifying linear relationships in the data. The confusion matrix for each model highlighted a high true positive rate, which is crucial in ensuring that most children with learning disabilities are correctly identified. The ROC-AUC scores further validated the robustness of the models, with Random Forest achieving the highest score, reflecting excellent discrimination capability. Feature importance analysis revealed that memory, attention, and language skills were among the most significant predictors. Visualization techniques such as heatmaps and bar plots helped in understanding the correlation between features and target labels. Cross-validation confirmed that the models were not overfitting and generalized well on unseen data. The study also showed that balanced datasets improved prediction quality, especially when using SMOTE. Overall, the model successfully automates LD detection and offers practical benefits to educators and psychologists. The integration of such intelligent systems can transform early intervention strategies. However, further improvements are possible with larger, more diverse datasets. Real-world testing in educational settings would also help refine and validate the system.



Conclusion

This project was developed with the aim of building a practical and intelligent system to assist in the early detection of learning disabilities in school-age children. As someone deeply interested in the application of technology for meaningful impact, I undertook this project to explore how machine learning can be used to solve real-world problems in education and child development. Through extensive data analysis, model training, and evaluation, I was able to develop a prediction system that leverages various algorithms—including Random Forest, Decision Tree, and Support Vector Machine—to accurately identify children who may be at risk of learning difficulties.
The results of this project not only demonstrated strong prediction accuracy but also showed how data-driven approaches can support teachers, parents, and psychologists in making informed decisions. By identifying key indicators such as memory, attention span, and language skills, the model provides valuable insights into the learning challenges faced by individual students.
This project has deepened my understanding of both machine learning techniques and their potential applications in the educational field. It also highlighted the importance of data quality, ethical considerations, and user-friendly interfaces in deploying such systems in real-world settings. Moving forward, I plan to improve the model by incorporating more diverse datasets and deploying it as a web-based tool for easier access and use. Overall, this project has been a meaningful step in combining my technical skills with a socially impactful goal.

REFERENCES

1. Chen, H., Fuller, S. S., Friedman, C., & Hersh, W. (2005). Knowledge Discovery in Data Mining and Text Mining in Medical Informatics. Medical Informatics, 3-34.

2 Cunningham, S. J., & Holmes, G. (1999). Developing innovative applications in agriculture using data mining. Proceedings of the Southeast Asia Regional Computer Confederation Conference.

3.David, J. M., & Kannan, B. (2009). Prediction of Frequent Signs of Learning Disabilities in School Age Children using Association Rules. Proceedings of the International Conference on Advanced Computing (ICAC 2009), 202-207.

4 David, J. M., & Kannan, B. (2010). Prediction of Learning Disabilities in School Age Children using Decision Tree. Communications in Computer and Information Science, 90(3), 533-542.

5. David, J. M., & Pramod, K. V. (2008). Prediction of Learning Disabilities in School Age Children using Data Mining Techniques. Proceedings of AICTE Sponsored National Conference on Recent Developments and Applications of Probability Theory, 139-146.

6 Han, J., & Kamber, M. (2008). Data Mining-Concepts and Techniques (2nd ed.). Morgan Kaufmann-Elsevier Publishers

.

7 Kothari, A., & Keskar, A. (2009). Rough Set Approach for Overall Performance Improvement of an Unsupervised ANN-Based Pattern Classifier. Journal on Advanced Computational Intelligence and Intelligent Information, 13(4), 434-440.

8 Palubinskas, G., Descombes, X., & Kruggel, F. (1998). An unsupervised clustering method using the entropy minimization. Proceedings of the Fourteenth International Conference on Pattern Recognition.

9 Roy, D. K., & Sharma, L. K. (2010). Genetic k-Means clustering algorithm for mixed numeric and categorical data sets. International Journal of Artificial Intelligence & Applications, 1(2), 23-28.