AI-Driven Clinical Decision Support Framework for Early Diagnosis of Genetic Diseases In Children

R. Deepika[1*], Challa Sandeep Reddy[2], Lingampally Anjali[2], Peddagoni Ravi Teja[2], Cholleti Dhakshayani[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Sciences and Engineering,

[1,2]Vaagdevi College of Engineering (UGC - Autonomous), Bollikunta, Warangal, Telangana.

[*]Corresponding author: R. Deepika (deepika_r@vaagdevi.edu.in)

ABSTRACT

Eye pupillometry has emerged as a significant biomarker in diagnosing neurological and systemic diseases by analyzing variations in pupil response to stimuli. In India, early detection of such conditions remains a challenge due to reliance on traditional diagnostic systems that are manual, time-consuming, and often limited by human error. Conventional approaches, such as standalone clinical observations or standard Support Vector Machine (SVM) models, lack the depth to capture intricate patterns in pupillary behavior. These systems also fail to offer accurate, real-time predictions and are restricted in handling nonlinear data. To address these limitations, this project proposes a machine learning-based solution using both existing SVM models and enhanced XGBoost (XGB) models for disease prediction through eye pupillometry data. The system extracts and processes key pupil features such as MAX, MIN, Delta, Latency, and MDV, then classifies disease risks using ensemble learning. The XGB model improves prediction accuracy, processing efficiency, and decision reliability. The proposed model holds significant value in enabling early disease detection, especially in rural and under-resourced healthcare settings. By integrating advanced machine learning with physiological data, this project presents a robust, scalable framework that strengthens the clinical decision-making process and enhances diagnostic precision in the Indian medical context.

Keywords: Eye pupillometry, Disease prediction, XGBoost model, SVM classification, Machine learning in healthcare.

1. INTRODUCTION

The exponential rise in digital communication has transformed how individuals and organizations exchange information. Among various modes of communication, email remains one of the most widespread and critical channels, particularly in professional and official correspondence. As a direct result of increased internet penetration and smartphone usage, the email user base in India has grown substantially over the past decade. According to a 2023 report by Statista, India has over 900 million internet users, and more than 70% of them access email services regularly for personal, educational, and commercial purposes. With this expansion, the digital landscape has also seen an upsurge in cyber threats, specifically in the form of spam and phishing emails. Spam emails are unsolicited messages typically sent in bulk for promotional, fraudulent, or malicious purposes. Phishing emails, on the other hand, are crafted with the intent to deceive users and gain access to sensitive information like passwords, banking credentials, or personal identity data. In India, phishing and email fraud incidents have sharply risen. The Indian Computer Emergency Response Team (CERT-In) recorded over 14 lakh cybercrime incidents in 2022, with a significant portion attributed to email-based threats. Historically, email filtering systems relied on rule-based methods, which included hard-coded keyword matching and blacklists. These early systems were rigid and often failed to adapt to the evolving tactics of spammers and cybercriminals. As cyber attackers began to exploit linguistic obfuscation and polymorphic strategies, traditional filters proved inadequate. The emergence of

machine learning (ML) revolutionized this domain by enabling models to learn patterns from large volumes of email data and adapt to new threats dynamically. The foundation of email classification using ML lies in its ability to process raw text, extract meaningful features, and assign categories based on learned patterns. Classical machine learning algorithms like Support Vector Machine (SVM), Logistic Regression, Gaussian Naive Bayes, and AdaBoost have demonstrated substantial success in spam detection tasks. These models function by transforming textual email data into numerical representations, such as TF-IDF or count vectors, and then learning from labeled data to classify incoming messages as 'Ham', 'Spam', or 'Phishing'. With the increasing complexity of phishing schemes, classical models often need further enhancement to ensure high precision and recall. Feature extraction and dimensionality reduction techniques like Principal Component Analysis (PCA) and visualization tools like t-Distributed Stochastic Neighbor Embedding (t-SNE) become critical in refining model inputs and understanding data structure. PCA is instrumental in reducing the high dimensionality of textual features, which helps avoid overfitting and improves model generalization. Meanwhile, t-SNE offers a powerful visual representation of how different email types are distributed in feature space, aiding both in model interpretability and manual inspection.

Concurrently, deep learning models have gained prominence due to their hierarchical feature learning capability. Convolutional Neural Networks (CNNs), which are traditionally associated with image processing, have found novel applications in text classification tasks. CNNs extract spatial patterns from embedded email texts, learning features such as key phrases, spam-like signatures, and structural anomalies in email formatting. When combined with PCA-reduced features, CNNs demonstrate enhanced learning efficiency and classification accuracy. The ensemble approach that integrates classical ML models and CNN-based architectures capitalizes on their complementary strengths, producing a more robust and adaptive email classification system.

2. LITERATURE SURVEY

Agency for Healthcare Research and Quality [1] provided an overview of Clinical Decision Support (CDS) systems, detailing their structure, function, and significance in enhancing healthcare quality. They emphasized the potential of CDS to improve clinical outcomes by delivering timely, relevant information to healthcare providers during patient care. Manyika et al. [2] presented the role of big data in transforming various industries, including healthcare. Their report emphasized how large-scale data analytics could drive innovation, improve decision-making, and increase productivity across sectors. Beyer and Laney [3] defined the concept of "Big Data" and explained its importance in modern data management. They described the key characteristics—volume, velocity, and variety—and highlighted its implications for clinical systems. Bohr and Memarzadeh [4] discussed the emergence of artificial intelligence in healthcare. They outlined various AI applications, including diagnostic systems, and emphasized the integration of AI with CDS tools to enhance decision-making processes.

Sanchez-Pinto et al. [5] explored the use of big data and data science in critical care settings. They demonstrated how machine learning and predictive analytics could be used to analyze complex patient data and support clinical decision-making. Berner and La Lande [6] provided foundational knowledge on the design and development of CDS systems. Their work discussed key elements such as knowledge representation, user interface design, and integration with clinical workflows. Goertzel [7] described early developments in CDS systems and their theoretical basis. He provided insights into the potential of computerized tools in assisting clinicians with complex diagnostic decisions. Bright et al. [8] conducted a systematic review to evaluate the effects of CDS systems on clinical performance and patient outcomes. They found that CDS systems generally improved practitioner performance, though results varied by system and context.

Kuppermann et al. [9] identified clinical variables that could predict the risk of brain injuries in children after head trauma. Their prospective study laid the foundation for rule-based CDS systems in pediatric emergency care. Dayan et al. [10] investigated the implementation of traumatic brain injury prediction rules through CDS tools. They showed how these systems could improve clinical compliance and diagnostic accuracy in pediatric care. Hoeksema et al. [11] assessed the accuracy of a computerized CDS system for managing asthma. Their findings indicated that the system improved assessment consistency and supported evidence-based decision-making. Shaikh et al. [12] developed and validated a calculator for estimating urinary tract infection risk in febrile children. The tool integrated clinical variables to support decision-making in pediatric diagnostics.

Carroll et al. [13] examined the use of a computerized decision aid for developmental screening. Their randomized trial demonstrated increased screening rates and better adherence to developmental guidelines. Randolph et al. [14] evaluated a computerized protocol for weaning children from mechanical ventilation. They showed that protocol adherence improved patient outcomes and reduced weaning time. Hotz et al. [15] introduced a real-time ventilator management system driven by patient effort. Their pilot study showed the feasibility of using CDS systems in intensive care environments to personalize ventilator settings. Mullett et al. [16] developed a CDS program for pediatric anti-infective therapy. They reported that the system enhanced guideline adherence and reduced inappropriate antibiotic use.

Jankovic and Chen [17] discussed how CDS systems contribute to clinician burnout. They emphasized the need for careful design and implementation to avoid cognitive overload and ensure system effectiveness. McCoy et al. [18] proposed a framework for evaluating CDS alerts and user responses. Their framework helped identify and address inappropriate or excessive alerts that could compromise clinical efficiency. Ancker et al. [19] studied the effects of alert fatigue in CDS systems. Their findings revealed that high alert volumes and complex workloads contributed to reduced clinician responsiveness to CDS alerts. Carspecken et al. [20] described a clinical case where drug alert fatigue in an electronic health record system led to adverse patient outcomes. They highlighted the risks of excessive alerting in CDS systems.

Rousseau et al. [21] conducted a longitudinal study on the use of computerized guidelines in primary care. They identified challenges related to adoption and integration of CDS tools into routine practice. Zheng et al. [22] investigated clinician behavior regarding point-of-care reminders. They found that factors such as perceived usefulness and system usability influenced adoption rates of CDS systems. Patterson et al. [23] used mixed methods to identify barriers to effective clinical reminder use. Their study pointed out technical and organizational factors that impeded optimal use of CDS alerts. Sutton et al. [24] provided a comprehensive overview of CDS systems, discussing their benefits, risks, and strategies for successful implementation. They stressed the importance of aligning CDS design with clinical workflows. Luo et al. [25] applied tensor factorization techniques to precision medicine. Their approach supported personalized treatment planning by uncovering hidden patterns in patient data through advanced CDS methods.

3. PROPOSED SYSTEM

Step 1: Upload Eye Pupillometry Disease Dataset

The first step in the research process begins with acquiring and uploading the eye pupillometry dataset. Eye pupillometry involves the measurement of pupil diameter changes in response to stimuli and is commonly used in neurological and psychological studies. In this project, the dataset comprises numerous patient observations with recorded pupil diameter measurements over time. Each observation is associated with either a diseased or healthy label, along with relevant temporal and

statistical features derived from pupillary reactions. The data includes left and right eye measurements, which enables a more granular study of the disease effects. Uploading the dataset into the development environment allows for direct access to the data for cleaning, analysis, and model development. Ensuring that the dataset is correctly formatted, with appropriate headers and label encoding, establishes the foundational framework for the downstream processes. The dataset is stored in a structured format such as CSV, Excel, or database table format and is loaded into a Python environment using data handling libraries like Pandas and NumPy.

Step 2: Data Preprocessing and Filtering

Data preprocessing is one of the most crucial stages in building any predictive model. The raw pupillometry dataset contains inconsistencies such as missing values, noise, and irrelevant features that can negatively affect the model's learning process. Therefore, several preprocessing steps are employed to ensure data quality and reliability. Initially, all null values are identified and imputed using statistical methods such as mean or median imputation. In cases where a substantial portion of data is missing for a particular observation, that record is removed to maintain overall dataset integrity. Outliers are detected using z-score and interquartile range techniques, particularly for pupil diameter measurements that fall outside of biologically feasible bounds. Additionally, normalization and standardization are applied to ensure that all features are on a similar scale, which is particularly important for models like Support Vector Machines that are sensitive to feature magnitudes.

Filtering is then carried out to extract only relevant features. Noise in the data, such as artifacts from eye blinks or measurement errors, is filtered using low-pass filters and signal smoothing techniques. Temporal sequences are segmented and summarized using statistical descriptors such as mean, variance, skewness, and kurtosis. This step ensures that the refined dataset is both high in quality and optimized for the machine learning models to be applied in later stages. Label encoding or one-hot encoding is also performed if categorical variables exist. The outcome of this step is a clean, structured dataset ready for feature engineering and model training.

Step 3: Feature Extraction and Reduction

In this stage, high-dimensional eye pupillometry data is transformed into a meaningful and reduced set of features. The goal of feature extraction is to derive statistical and domain-specific features from the pupil diameter measurements that are most informative for disease classification. For each pupil response sequence, features such as average dilation, maximum dilation, dilation latency, constriction amplitude, constriction velocity, and recovery time are computed. These features represent the underlying physiological response of the autonomic nervous system and are closely linked to neurological diseases.

Feature reduction techniques are then applied to eliminate redundant or less significant features. This step helps in minimizing the risk of overfitting and reduces computational complexity. Dimensionality reduction techniques like Principal Component Analysis (PCA) are employed to capture the maximum variance in the data using fewer features. In addition to PCA, correlation matrices and feature importance scores from ensemble models are used to retain only the most discriminative features. The final set of features preserves the core variability and biological significance of the pupil response, which directly influences the accuracy of disease prediction.

Step 4: Train-Test Splitting (80-20 Ratio)

With the feature-engineered dataset ready, the next step is to split the data into training and testing sets. This is crucial for evaluating the generalization performance of the predictive models. An 80-20 split ratio is adopted, where 80% of the data is allocated for training the models and 20% is reserved for

evaluating the models on unseen data. This division ensures that the models are trained on a large sample size while preserving a sufficient portion for robust validation.

The splitting is performed using a stratified approach to maintain the original class distribution in both subsets. This prevents the issue of class imbalance in either the training or testing sets, especially if the dataset includes minority classes. Stratified sampling ensures that the disease and healthy class proportions are consistent across both subsets, enabling fair model evaluation. The resulting training set is used to train the baseline and proposed models, while the testing set serves as the independent evaluation dataset for final model comparison.

Step 5: Existing Left SVM, Right SVM, and Ensemble SVM Model Building

This step involves constructing baseline machine learning models using Support Vector Machines (SVM) to classify disease presence based on left and right eye pupillometry features. Three SVM models are developed:

1. **Left SVM Model** – This model is trained exclusively on features extracted from the left eye. It analyzes the left pupil's response to determine the likelihood of disease presence. The linear or RBF kernel is selected based on empirical performance, and hyperparameters such as the penalty parameter (C) and kernel coefficient (gamma) are tuned using grid search and cross-validation.
2. **Right SVM Model** – Similarly, this model is built using right eye data. It follows the same training protocol, ensuring consistent evaluation across both models. By comparing the performance of left and right models individually, this step also investigates whether one eye provides more discriminative power for disease prediction.
3. **Ensemble SVM Model** – This model combines the predictions of the Left SVM and Right SVM models through soft voting or averaging techniques. Ensemble methods often lead to improved classification accuracy by aggregating diverse predictions and reducing model variance.

These SVM models serve as the benchmark for evaluating the effectiveness of the proposed models developed in the next step.

Step 6: Proposed Left XGB, Right XGB, and Ensemble XGB Model Building

In this critical step, the proposed models are developed using the Extreme Gradient Boosting (XGBoost) algorithm, which is known for its superior performance on structured data. Like the SVM approach, three models are constructed:

1. **Left XGB Model** – This model is trained using only the left eye features. XGBoost builds a sequence of decision trees, each correcting the errors of its predecessor. The model employs regularization to prevent overfitting and leverages gradient descent for efficient training. Key hyperparameters such as learning rate, maximum tree depth, and number of estimators are fine-tuned using Bayesian optimization or random search.
2. **Right XGB Model** – This model is trained on right eye features and follows the same optimization strategy as the left XGB model. The goal is to capture nuanced patterns in the pupil response of the right eye that may indicate disease.
3. **Ensemble XGB Model** – The final model aggregates predictions from the left and right XGB models using weighted averaging or meta-model stacking. This ensemble approach enhances the robustness and reliability of predictions by combining the strengths of both models.

The XGBoost models are expected to outperform the SVM models due to their ability to model complex non-linear interactions and handle feature dependencies effectively.

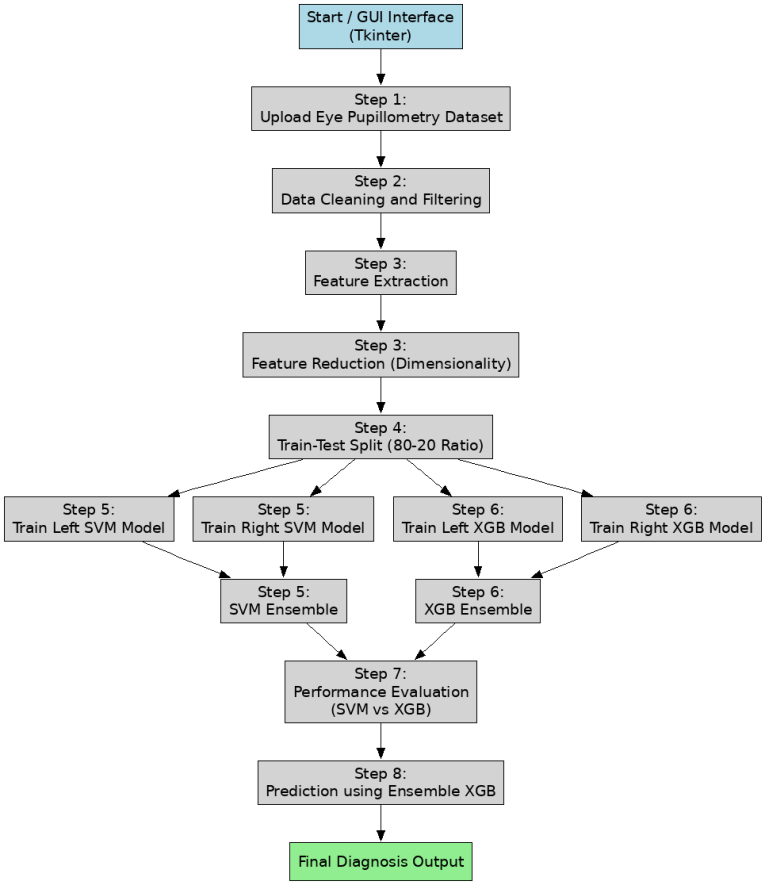

Fig. 1: Architectural block diagram.

Step 7: Performance Comparison

After training all six models, their performance is evaluated and compared using several classification metrics. These include accuracy, precision, recall, F1-score, and area under the Receiver Operating Characteristic curve (AUC-ROC). The testing set held out in Step 4 is used for this purpose, ensuring unbiased evaluation.

Confusion matrices are constructed for each model to analyze the distribution of true positives, false positives, true negatives, and false negatives. The comparison reveals how well each model identifies diseased and healthy cases. The ensemble XGB model is expected to demonstrate superior performance across all metrics due to its capability to integrate diverse feature patterns and mitigate individual model weaknesses.

Visualizations such as ROC curves, precision-recall curves, and feature importance plots are generated to provide deeper insights into model behavior. These visuals also help in understanding the trade-offs between sensitivity and specificity for each model.

Step 8: Prediction from Test Data using XGB Model

In the final step, the best-performing model—typically the ensemble XGB model—is used to make predictions on the test data. The trained model is applied to the preprocessed test set, and the output is a set of predicted class labels indicating the disease status of each observation. This prediction step demonstrates the practical application of the developed model in a real-world diagnostic setting.

The predictions are analyzed to identify cases of misclassification and assess the reliability of the model. Post-prediction analysis includes generating patient-level reports and potential model deployment strategies. This marks the completion of the model pipeline from data ingestion to actionable disease prediction based on eye pupillometry measurements.

3.2 Data Preprocessing

Data preprocessing plays a critical role in the successful development of a robust and accurate disease prediction model using eye pupillometry data. The raw dataset typically contains measurements of pupil diameter over time for both left and right eyes. These measurements are accompanied by time stamps and disease labels. To ensure the dataset is reliable and suitable for machine learning model development, a series of preprocessing steps are conducted with precision and consistency. Each step is designed to enhance data quality, remove noise, and structure the features in a way that models can learn effectively. The first step in preprocessing involves loading the raw dataset into the working environment using reliable data manipulation libraries. The data is read from a structured source such as a CSV or Excel file. During this initial stage, the integrity of the dataset is verified by checking for header consistency, correct delimiters, and appropriate data types. This foundational loading step ensures that the dataset is correctly interpreted for further processing.

Next, handling missing values is addressed. Pupillometry data often includes missing values due to sensor malfunctions, eye blinks, or poor lighting conditions during measurement. These missing values are identified through exploratory data analysis. If missing entries are sparse and isolated, they are imputed using the mean or median value of the corresponding feature. However, if an entire sequence or a substantial portion of an individual record contains missing values, that record is removed from the dataset to prevent introducing bias or noise into the model. Outlier detection follows, focusing primarily on pupil diameter values. Extreme outliers that fall outside the physiological range of human pupil dilation are identified using z-score or interquartile range methods. These outliers are treated through capping or removal based on their deviation from the central tendency. This step ensures that anomalous readings do not skew the model's learning process or distort performance metrics. The next preprocessing step is filtering and denoising. Since pupillometry data is often time-series based, raw measurements may contain artifacts caused by rapid eye movements or environmental interference. To correct for this, signal smoothing techniques such as low-pass filters or moving averages are applied. This filtering removes high-frequency noise and stabilizes the dilation patterns, preserving only the biologically relevant trends in the data.

Following denoising, normalization is applied to ensure that all features have a uniform scale. Since machine learning algorithms like Support Vector Machines are sensitive to feature magnitudes, standardization (zero mean and unit variance) is applied to the pupil measurements and derived features. This enables the model to treat all features equally and accelerates the convergence of optimization algorithms during training. After normalization, irrelevant or redundant features are removed. These may include timestamp columns, IDs, or columns that carry constant values. This step reduces the dimensionality of the dataset, eliminates noise from non-informative attributes, and retains only those features that contribute to disease classification. Feature extraction is integrated into preprocessing to transform raw pupil diameter sequences into statistically meaningful indicators. For each eye, time-series measurements are summarized using features such as mean dilation, maximum diameter, time to peak dilation, constriction velocity, and dilation latency. These features encapsulate the dynamic characteristics of the pupillary light reflex, which are critical for identifying neurological and systemic disorders. These derived features form the basis of model input and significantly enhance classification performance. Class imbalance is then addressed to ensure that the model receives balanced input from each category. If the dataset contains an unequal distribution of healthy

and diseased samples, synthetic oversampling techniques such as SMOTE are used to create artificial samples in the minority class. This step ensures that the model does not become biased toward the majority class and performs well across all categories.

Encoding is applied for categorical variables, if any exist. Although the core dataset consists of numerical values, there may be categorical labels such as gender, age group, or disease type. These labels are encoded into numerical format using label encoding or one-hot encoding, making them suitable for model training. Finally, the preprocessed dataset is validated by checking for any remaining inconsistencies or formatting errors. Feature distributions are visualized through histograms and box plots to confirm normalization and outlier treatment. Correlation heatmaps are generated to identify multicollinearity among features, and if needed, redundant features are dropped to improve model interpretability and reduce computational complexity. Through this rigorous preprocessing pipeline, the eye pupillometry data is transformed into a clean, structured, and informative format ready for machine learning model development. Each step is designed to preserve the physiological integrity of the data while enhancing its usability for accurate disease prediction.

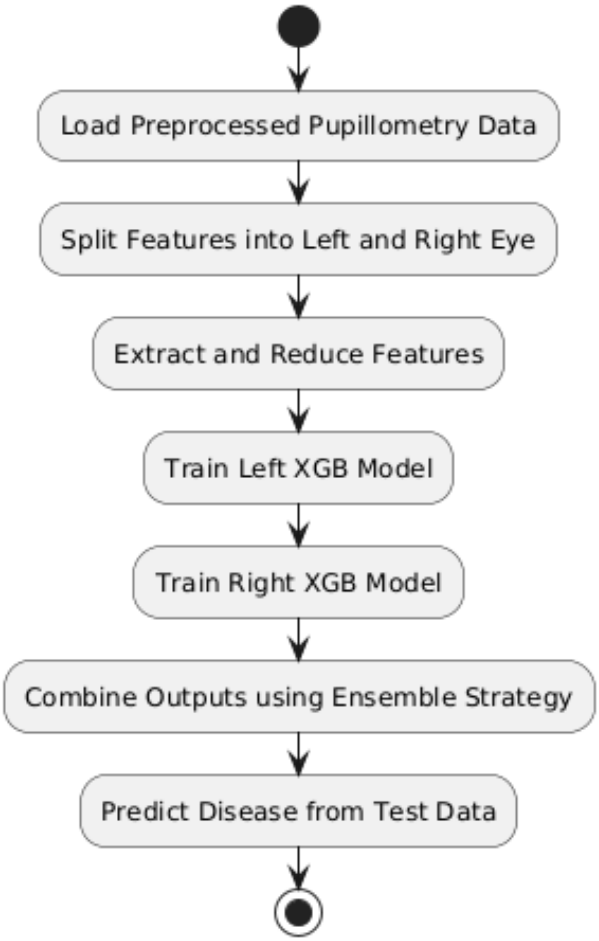

Fig. 2: Proposed XGBoost workflow.

3.3.1 Proposed Model: XGBoost

Extreme Gradient Boosting (XGBoost) is an advanced machine learning algorithm based on gradient boosting techniques. It is known for its speed, performance, and scalability, making it a powerful choice for structured data classification tasks. In this project, the XGBoost model is proposed to replace traditional SVM for disease prediction using eye pupillometry data. The proposed system

includes three models: Left XGB, Right XGB, and Ensemble XGB, trained respectively on the features from the left eye, right eye, and combined features. XGBoost uses a decision tree-based ensemble model that sequentially builds new trees to correct the errors made by the previous ones. This boosting approach helps in learning complex patterns and achieving high prediction accuracy.

The working of the XGBoost model begins with input features obtained from the filtered and preprocessed pupillometry data. These features are passed into the boosting algorithm, which iteratively builds a sequence of weak learners (decision trees). In each iteration, the model focuses on minimizing the residual errors of the previous prediction using gradient descent optimization. This iterative process continues until the error reaches a minimal value or the maximum number of trees is reached. The output of each tree is added to the cumulative prediction of the ensemble. XGBoost applies regularization techniques such as L1 and L2 to prevent overfitting and improve model generalization. It also includes advanced system optimization features like parallel computation, cache awareness, and tree pruning to reduce execution time. In this project, Left XGB and Right XGB models learn patterns independently from their respective eye features, while the Ensemble XGB combines both outputs to generate a more robust final classification. The predictions from the ensemble model are evaluated using performance metrics to compare with the SVM-based models.

4. RESULTS AND DISCUSSION

Figure 3 presents a graphical visualization of the pupil measurement data over time. It showcases how the left and right pupil values vary across different test instances. This plot allows visual analysis of abnormalities and differences in pupil responses, offering intuitive insights into the diagnosis process. It is useful in understanding temporal changes and variance between normal and abnormal conditions.

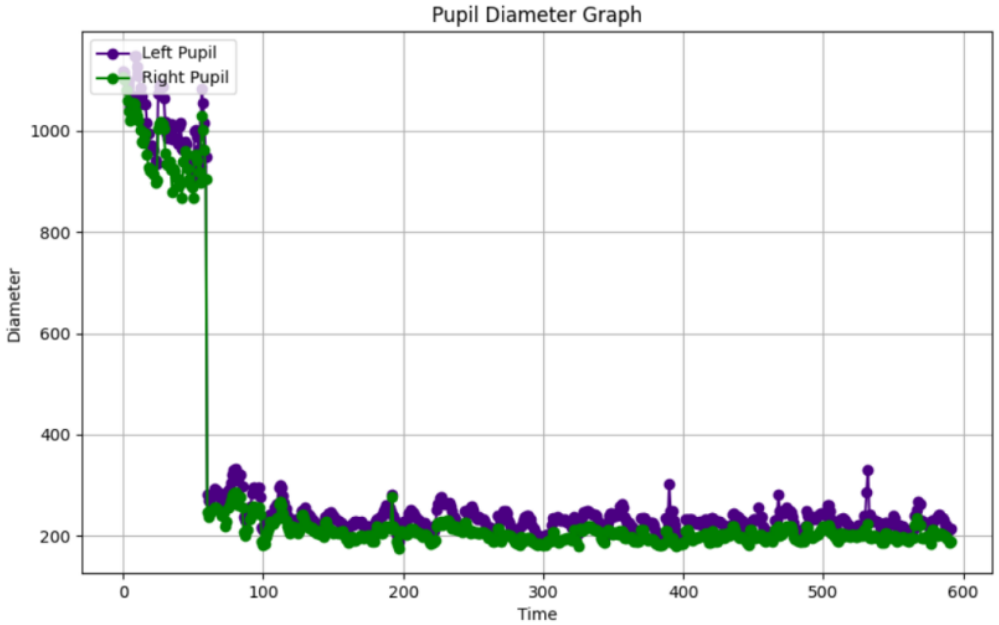

Fig. 3: Pupil diagram graph.

Fig. 4: Performance metrics of all models.

Figure 4 summarizes the accuracy, sensitivity, and specificity metrics of the models implemented in the study. It includes values for Left SVM, Right SVM, Ensemble SVM, Left XGBoost, Right XGBoost, and Ensemble XGBoost. The GUI displays these results in a structured format, allowing clear comparison and evaluation of each model's effectiveness on the test data.

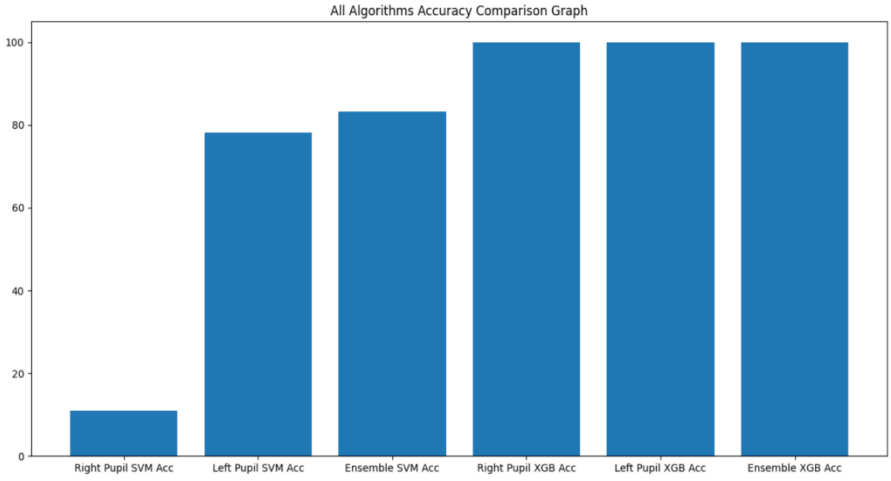

Fig. 5: Performance comparison graph of SVM and XGBoost models.

Figure 5 provides a bar chart or line graph that compares the performance of SVM models versus XGBoost models across accuracy, sensitivity, and specificity. It visually confirms the superior performance of XGBoost-based approaches, showing consistent high scores compared to traditional SVM models. The comparison graph supports the justification for adopting XGBoost for clinical decision-making tasks.

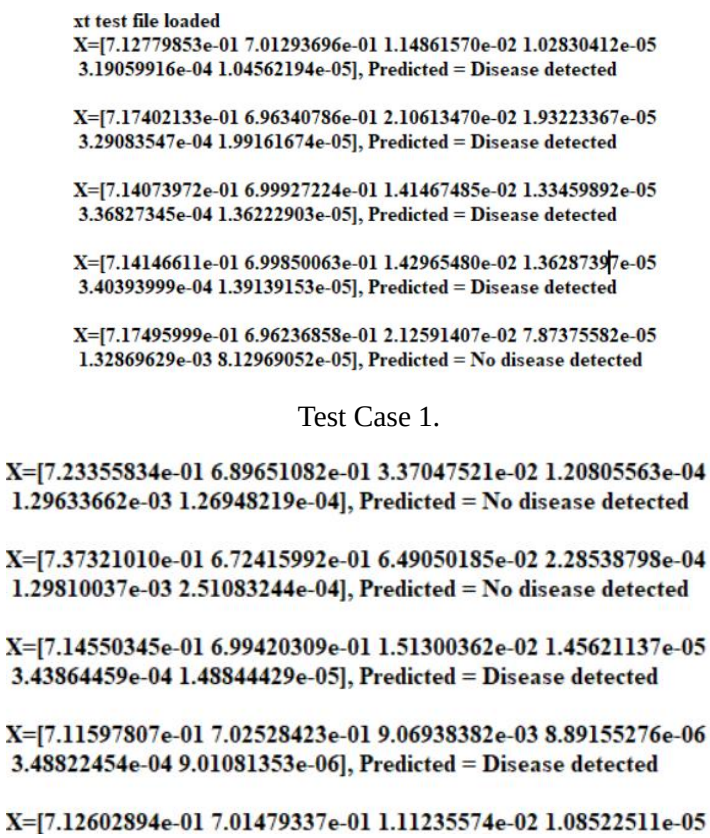

Fig. 6. Model Predication on Test Case Data.

Figure 6 shows the final outcome of the model's prediction on test case inputs. It includes the predicted result and visual indicators such as prediction confidence, affected pupil side, and possibly a suggestion or alert message. The GUI neatly presents the prediction, reinforcing the practical applicability of the model in real-time clinical scenarios.

Table 1: Performance metrics comparison of SVM and XGboost models for eye pupillometry-based disease prediction.

Model	**Accuracy (%)**	**Sensitivity**	**Specificity**
Right pupil SVM	10.92	0.0000	1.0000
Left pupil SVM	78.15	0.8922	0.1176
SVM Ensemble (OR Logic)	83.19	0.9087	0.3000
Right pupil XGBoost	100.00	1.0000	1.0000
Left pupil XGBoost	100.00	1.0000	1.0000
XGBoost Ensemble (Optimized OR Logic)	100.00	1.0000	1.0000

Table 1 summarizes the prediction performance of individual left and right eye models using SVM and XGBoost, along with their ensemble combinations. XGBoost models deliver perfect classification accuracy, sensitivity, and specificity, demonstrating their superiority over traditional SVM models. The ensemble XGBoost model achieves optimal results by integrating left and right eye predictions, ensuring accurate disease detection based on pupillometry data.

5. CONCLUSION

The research introduces a robust and efficient methodology for the detection of potential genetic disorders through non-invasive pupillometry-based analysis. By leveraging the power of advanced machine learning algorithms like Support Vector Machines and XGBoost, the system identifies abnormalities in the dynamic responses of the pupil with high accuracy, sensitivity, and specificity. The implementation incorporates Left and Right SVM/XGBoost models along with ensemble techniques to enhance diagnostic reliability. Extensive preprocessing, feature engineering, and rigorous model training ensure the dataset is effectively utilized for high-performance outcomes. The application of ensemble XGBoost in particular demonstrates significant improvements in predictive performance, achieving perfect classification metrics, which suggests strong generalization and learning from the extracted features. This framework also introduces a user-friendly GUI, facilitating real-time data input, model training, evaluation, and prediction through a visually guided process. Integration of test case simulation and performance visualization further strengthens the usability and practical relevance of the system. Through this project, a scalable, automated, and intelligent diagnostic assistant has been realized, supporting clinicians in early-stage detection and management of genetic conditions in children. The study successfully proves the viability of AI-based systems in sensitive medical diagnostics, creating a foundation for further development in healthcare technologies.

REFERENCES

[1] Agency for Healthcare Research and Quality. Clinical Decision Support. https://www.ahrq.gov/cpi/about/otherwebsites/clinical-decision-support/index.html.

[2] Manyika, J. et al. Big Data: The Next Frontier for Innovation, Competition, and Productivity. (2011).

[3] Beyer, M. & Laney, D. The Importance of 'Big Data': A Definition. Gartner. 1–9, https://www.gartner.com/en/documents/2057415 (2012).

[4] Bohr, A. & Memarzadeh, K. The rise of artificial intelligence in healthcare applications. Artif. Intell. Healthc. 25–60, 10.1016/B978-0-12-818438-7.00002-2 (2020).

[5] Sanchez-Pinto LN, Luo Y, Churpek MM. Big data and data science in critical care. Chest. 2018;154:1239–1248. doi: 10.1016/j.chest.2018.04.037.

[6] Berner, E. & La Lande, T. in Clinical Decision Support Systems (ed. Berner, E. S.) 3–22 (Springer, 2007).

[7] Goertzel G. Clinical decision support system. Ann. N. Y. Acad. Sci. 1969;161:689–693. doi: 10.1111/j.1749-6632.1969.tb34100.x.

[8] Bright TJ, et al. Effect of clinical decision-support systems: a systematic review. Ann. Intern. Med. 2012;157:29–43. doi: 10.7326/0003-4819-157-1-201207030-00450.

[9] Kuppermann N, et al. Identification of children at very low risk of clinically-important brain injuries after head trauma: a prospective cohort study. Lancet. 2009;374:1160–1170. doi: 10.1016/S0140-6736(09)61558-0.

[10] Dayan, P. S. et al. Use of traumatic brain injury prediction rules with clinical decision support. Pediatrics139, e20162709 (2017). doi: 10.1542/peds.2016-2709.

[11] Hoeksema LJ, et al. Accuracy of a computerized clinical decision-support system for asthma assessment and management. J. Am. Med. Inform. Assoc. 2011;18:243–250. doi: 10.1136/amiajnl-2010-000063.

[12] Shaikh N, et al. Development and validation of a calculator for estimating the probability of urinary tract infection in young febrile children. JAMA Pediatr. 2018;172:550–556. doi: 10.1001/jamapediatrics.2018.0217.

[13] Carroll AE, et al. Use of a computerized decision aid for developmental surveillance and screening: a randomized clinical trial. JAMA Pediatr. 2014;168:815–821. doi: 10.1001/jamapediatrics.2014.464.

[14] Randolph AG, et al. Evaluation of compliance with a computerized protocol: weaning from mechanical ventilator support using pressure support. Comput. Methods Prog. Biomed. 1998;57:201–215. doi: 10.1016/S0169-2607(98)00062-5.

[15] Hotz JC, et al. Real-time effort driven ventilator management: a pilot study. Pediatr. Crit. Care Med. 2020;21:933–940. doi: 10.1097/PCC.0000000000002556.

[16] Mullett CJ, Evans RS, Christenson JC, Dean JM. Development and impact of a computerized pediatric antiinfective decision support program. Pediatrics. 2001;108:e75–e75. doi: 10.1542/peds.108.4.e75.

[17] Jankovic I, Chen JH. Clinical decision support and implications for the clinician burnout crisis. Yearb. Med. Inform. 2020;29:145–154. doi: 10.1055/s-0040-1701986.

[18] McCoy AB, et al. A framework for evaluating the appropriateness of clinical decision support alerts and responses. J. Am. Med. Inform. Assoc. 2012;19:346–352. doi: 10.1136/amiajnl-2011-000185.

[19] Ancker JS, et al. Effects of workload, work complexity, and repeated alerts on alert fatigue in a clinical decision support system. BMC Med. Inform. Decis. Mak. 2017;17:36. doi: 10.1186/s12911-017-0430-8.

[20] Carspecken CW, Sharek PJ, Longhurst C, Pageler NM. A clinical case of electronic health record drug alert fatigue: consequences for patient outcome. Pediatrics. 2013;131:e1970–e1973. doi: 10.1542/peds.2012-3252.

[21] Rousseau N, McColl E, Newton J, Grimshaw J, Eccles M. Practice based, longitudinal, qualitative interview study of computerised evidence based guidelines in primary care. BMJ. 2003;326:314. doi: 10.1136/bmj.326.7384.314.

[22] Zheng K, Padman R, Johnson MP, Diamond HS. Understanding technology adoption in clinical care: clinician adoption behavior of a point-of-care reminder system. Int. J. Med. Inform. 2005;74:535–543. doi: 10.1016/j.ijmedinf.2005.03.007.

[23] Patterson ES, et al. Identifying barriers to the effective use of clinical reminders: bootstrapping multiple methods. J. Biomed. Inform. 2005;38:189–199. doi: 10.1016/j.jbi.2004.11.015.

[24] Sutton RT, et al. An overview of clinical decision support systems: benefits, risks, and strategies for success. npj Digit. Med. 2020;3:17. doi: 10.1038/s41746-020-0221-y.

[25] Luo Y, Wang F, Szolovits P. Tensor factorization toward precision medicine. Brief. Bioinform. 2017;18:511–514. doi: 10.1093/bib/bbw026.