

Arcus: A Data-Driven Web-Based Autism Screening System Using Machine Learning on Survey-Derived Behavioral Features

Mridul Narain
Computing Technologies, SRM
Institute of Science and Technology,
SRM Nagar, Kattankulathur,
Tamilnadu, India.

Suyash Shinde
Computing Technologies SRM
Institute of Science and Technology,
SRM Nagar, Kattankulathur,
Tamilnadu, India.

Dr. Mangalagowri R
Computing Technologies, SRM
Institute of Science and Technology,
SRM Nagar, Kattankulathur,
Tamilnadu, India.

Abstract— Diagnosing Autism Spectrum Disorder (ASD) at an early stage is hindered by a shortage of specialists, lengthy clinical procedures, and limited accessibility in underserved regions. While the AQ-10 behavioral questionnaire offers a structured basis for preliminary screening, its manual administration and interpretation leave considerable room for automation. This paper presents a machine learning pipeline trained on the UCI Autism Screening Adult dataset, comprising 704 adult records with ten behavioral indicators and associated demographic attributes, to classify ASD likelihood with high accuracy. Six supervised classifiers were benchmarked: Decision Tree, Random Forest, K-Nearest Neighbors, Naïve Bayes, Gradient Boosting, and XGBoost. Random Forest emerged as the best-performing model, achieving 99.29% accuracy, an F1-score of 98.59%, and a mean 5-fold cross-validation accuracy of 95.88%. The trained model was deployed within Arcus, a full-stack web application built on React, Node.js, and PostgreSQL, offering secure user authentication, a guided AQ-10 questionnaire, real-time predictions with confidence scoring, clinical recommendations, and longitudinal result tracking. The system demonstrates that ensemble-based screening, when paired with a thoughtfully designed clinical interface, can meaningfully lower the barrier to early ASD identification.

Keywords: Autism Spectrum Disorder, AQ-10, machine learning, Random Forest, ensemble learning, web application, clinical decision support

I. INTRODUCTION

The growing prevalence of Autism Spectrum Disorder has placed mounting pressure on healthcare systems worldwide to develop faster, more accessible pathways to early identification. ASD is a lifelong neurodevelopmental condition marked by challenges in social communication, behavioral flexibility, and sensory processing. The Centers for Disease Control and Prevention estimates that approximately 1 in 36 children in the United States are diagnosed with ASD, and a non-trivial portion of affected individuals, particularly those with subtler presentations, go undiagnosed well into adulthood. This diagnostic gap carries real consequences: delayed intervention narrows the window during which behavioral and cognitive therapies yield the most measurable benefit.

Formal diagnosis of ASD typically involves multi-session clinical evaluations, standardized instruments such as the Autism Diagnostic Observation Schedule (ADOS-2), and input from multidisciplinary teams. While thorough, this pathway is resource-intensive and geographically uneven in availability. In many low- and middle-income regions, access to developmental specialists remains severely limited,

leaving families without actionable guidance for extended periods. Even in well-resourced settings, waiting times between initial concern and confirmed diagnosis can stretch across months or years. There is, therefore, a clear and practical motivation for screening tools that can serve as a reliable first filter, not as a replacement for clinical diagnosis, but as a means of identifying individuals who warrant urgent referral.

The AQ-10, a ten-item behavioral self-report instrument derived from the Autism Quotient scale, has been clinically validated as a brief screening tool for adults. Its binary response format and strong discriminative properties make it a natural candidate for machine learning-based classification. Several prior studies have explored this direction, applying classifiers such as decision trees, support vector machines, and neural networks to AQ-10-derived datasets. However, most of this work remains confined to isolated experimental pipelines with no accompanying deployment, the gap between a trained model and a usable clinical tool rarely gets addressed.

This paper closes that gap. We present a full end-to-end system consisting of two tightly coupled components: a comparative machine learning study across six classifiers and a production-grade web application named *Arcus* that operationalizes the best-performing model. The dataset used is the UCI Autism Screening Adult dataset, containing 704 adult records with AQ-10 behavioral scores and demographic metadata. Among the six algorithms evaluated, Decision Tree, Random Forest, K-Nearest Neighbors, Naïve Bayes, Gradient Boosting, and XGBoost, Random Forest delivered the strongest results, with 99.29% accuracy, an F1-score of 98.59%, and a 5-fold cross-validation mean of 95.88%. This model was subsequently integrated into Arcus, a full-stack application built on React, Node.js, and PostgreSQL, providing authenticated users with a guided questionnaire, real-time predictions, confidence scoring, and longitudinal result tracking.

The primary contributions of this work are as follows:

- To benchmark six supervised learning algorithms on the UCI Autism Screening Adult dataset using accuracy, F1-score, and cross-validation mean, and to identify the optimal classifier for deployment.
- To develop *Arcus*, a secure, full-stack ASD screening web application that integrates the trained Random Forest model with a clinically

informed user interface supporting prediction, recommendation, and history tracking.

- To demonstrate that an end-to-end machine learning system, spanning data preprocessing, model selection, and web deployment, can serve as a scalable and accessible preliminary screening tool for ASD in adult populations.

The remainder of this paper is organized as follows. Section II surveys related work in machine learning-based ASD screening. Section III describes the dataset and its characteristics. Section IV details the preprocessing pipeline, model training, and evaluation methodology. Section V presents experimental results and comparisons. Section VI describes the architecture and design of the Arcus web application. Section VII discusses findings, limitations, and future directions. Section VIII concludes the paper.

II. RELATED WORK

Machine learning applications in ASD detection have expanded considerably over the past decade, spanning behavioral questionnaire data, neuroimaging signals, physiological biomarkers, and hybrid approaches. This section reviews the most relevant prior work and identifies the gaps that motivated this study.

Questionnaire-Based Classification. The AQ-10 dataset family, covering toddler, child, adolescent, and adult cohorts, has become the de facto benchmark in this space. Hossain et al. [11] compared MLP, SVM, and Random Forest across all four age-group datasets and reported that MLP paired with Relief-F feature selection achieved 100% accuracy, suggesting that feature selection can matter more than model choice on this data. Hasan et al. [3] extended this by systematically testing eight classifiers across four scaling strategies, finding that AdaBoost performed best on toddler and child cohorts while LDA led on adolescent and adult data, a result that underscores the sensitivity of model performance to preprocessing choices. Vakadkar et al. [10] found Logistic Regression outperforming ensemble methods on the Q-CHAT-10 children's dataset, which is consistent with the dataset's small size and near-linear separability. Thabtah and Peebles [13] proposed a Rules-Machine Learning (RML) method that generates interpretable rule sets alongside predictions, outperforming boosting and bagging approaches across multiple ASD datasets, a notable result given the clinical value of explainability. Parpinelli et al. [7] took a different angle by training on healthcare provider records rather than questionnaire responses, achieving 73.62% accuracy with a stacking ensemble, demonstrating that screening is feasible even without formal questionnaire data.

Neuroimaging and Physiological Signal Approaches. Wall et al. [15] showed that just 8 of 29 ADOS items were sufficient for near-perfect autism classification, validated across three independent datasets, a result that highlighted the redundancy in standard diagnostic instruments and the potential for ML to identify high-signal features. Yaneva et al. [5] used eye-tracking data collected during web browsing to classify high-functioning autism in adults with approximately 74% accuracy, opening a non-invasive, non-self-report screening pathway. Saranya and Menaka [8] applied quantum SVM to EEG signals from a small cohort of children, reaching 98.9% accuracy with an augmented feature set, demonstrating that even minimal electrode

configurations can yield strong classification performance when features are carefully selected. Briguglio et al. [9] applied regression and ensemble methods to retrospective clinical data alongside ADOS-2 scores, finding gut disturbances, sleep problems, and EEG retrievals to be the strongest predictors of social affect scores.

Biological and Hybrid Approaches. Pinto et al. [2] explored FTIR spectroscopy on saliva samples as a non-behavioral ASD biomarker, with SVM achieving 92% sensitivity and 95% specificity. Shapley analysis identified specific protein spectral regions as the most discriminating features, suggesting a potential complement to questionnaire-based tools in cases where self-report is unreliable. Kavadi et al. [6] proposed a large-scale hybrid model combining a modified Squirrel Search Algorithm for feature selection with an Autoencoder-Butterfly Optimization classifier under a MapReduce infrastructure, reporting 92% accuracy across three ASD datasets, notable for being designed to scale to population-level data volumes.

Review and Methodological Work. Hyde et al. [14] reviewed 45 supervised ML papers in ASD research and found SVM and Random Forest to be the most consistently strong performers, while flagging widespread issues with small samples and inadequate cross-validation reporting. Kohli et al. [4] reviewed 35 technology-based detection studies focused on children under six, noting that deep learning models could detect ASD risk in infants as young as 9-12 months but cautioning against cultural bias and poor generalization outside lab settings. Mertz [1] surveyed deployed AI tools including FDA-authorized Canvas Dx, which combines questionnaire data with parent-submitted video, situating the field relative to regulatory and clinical deployment standards. Bone et al. [12] attempted to reproduce two earlier high-profile ML studies and failed on a larger, more balanced dataset, tracing the inflation to test data leakage, class imbalance mishandling, and over-reliance on raw accuracy. Their recommended use of unweighted average recall as the primary metric for imbalanced clinical datasets has since influenced best practice in the field.

Identified Gaps. Several limitations emerge consistently across this body of work. First, almost no study produces a deployable user-facing tool, the pipeline ends at model evaluation. Second, adult screening is underserved relative to pediatric cohorts. Third, single-model studies make it difficult to assess whether reported performance reflects the algorithm or the dataset. Fourth, most high-accuracy models offer no explanation alongside the prediction. Fifth, none of the reviewed ML papers address authentication, data protection, or responsible handling of health-related user data. This work directly addresses all five gaps through a six-classifier comparative study, adult-focused dataset, full-stack deployment, confidence-scored predictions with plain-language explanations, and JWT-secured data handling.

III. MATERIALS AND METHOD

The system functions in accordance with the procedure shown in Fig. 1.

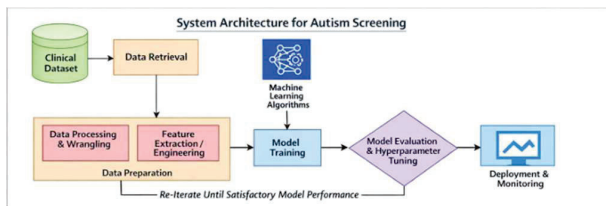


Fig 1. Proposed ASD screening architecture diagram.

A. Hardware Environment

TABLE I. HARDWARE ENVIRONMENT

GPU	Intel(R) UHD Graphics
CPU	12 th Gen Intel® Core™ i7-12650H
Installed Memory(RAM)	16GB
Operating System	Windows 11 (64-bit OS)

B. Software Environment

Model training and experimentation were carried out in a **Kaggle notebook environment** using **Python 3.11**, with **Scikit-learn** handling classifier implementation, **Pandas** and **NumPy** managing data preprocessing, and **Joblib** serializing the final trained model. The web application was built as a monorepo using **pnpm workspaces**, with a **React + Vite (TypeScript)** frontend styled via **Tailwind CSS** and a **Node.js + Express** backend. Data persistence was handled by **PostgreSQL** through **Drizzle ORM**, and user sessions were managed using **JWT-based authentication**.

C. Feature Preprocessing and Encoding

The UCI Autism Screening Adult dataset was preprocessed before model training. Numeric columns containing missing entries, primarily age, were imputed using the column median, while categorical attributes such as gender, ethnicity, and relation were imputed using the mode. All categorical variables were subsequently converted to numeric representations through one-hot encoding using Pandas' `get_dummies()` function. The cumulative behavioral score column was dropped prior to training to prevent data leakage, since it is a direct arithmetic sum of the ten AQ-10 responses and would otherwise trivially determine the target label. The column structure generated during training was saved separately and used at inference time to align incoming user inputs to the same feature space, ensuring that the deployed prediction endpoint operated under conditions identical to those used during model evaluation.

D. Machine Learning Algorithm

The screening system implements and evaluates six supervised learning algorithms to classify individuals as ASD Positive or ASD Negative. Each record in the training data consists of a feature vector $x_i \in \mathbb{R}^d$ and a binary class label $y_i \in \{0, 1\}$. The six classifiers, **Decision Tree (DT)**, **Random Forest (RF)**, **K-Nearest Neighbors (KNN)**, **Naïve Bayes (NB)**, **Gradient Boosting (GB)**, and **Extreme Gradient Boosting (XGBoost)**, were selected to cover a range of learning paradigms, from simple rule-based splitting to complex ensemble methods, allowing a meaningful performance comparison across fundamentally different approaches.

Decision Tree recursively partitions the feature space by selecting the split at each node that maximizes information gain, producing an interpretable tree structure. While fast to

train, single trees are prone to overfitting on small datasets.

K-Nearest Neighbors classifies a sample by identifying the k most similar training instances in feature space and assigning the majority class label. For this study, k was tuned via GridSearchCV over $\{3, 5, 7, 9\}$ with 5-fold cross-validation, yielding an optimal $k = 7$. KNN is non-parametric but sensitive to feature scaling, which was handled through a StandardScaler pipeline.

Naïve Bayes applies Bayes' theorem under the assumption of conditional feature independence, estimating the posterior probability of each class given the input features. Its simplicity makes it computationally efficient but potentially limiting on datasets where feature correlations exist.

Gradient Boosting builds an additive ensemble of weak learners sequentially, with each tree trained to correct the residual errors of the previous one. It tends to generalize well but is more sensitive to hyperparameter choices than bagging-based methods.

XGBoost extends Gradient Boosting with regularization terms added directly to the objective function, second-order gradient approximations, and built-in handling of missing values, making it more efficient and less prone to overfitting than standard Gradient Boosting on tabular data.

Among all six, **Random Forest** was selected as the deployment model based on its performance across all three evaluation metrics. A detailed breakdown of its algorithmic procedure follows.

Step 1: Data Input and Bootstrap Sampling

Given a training dataset $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ of n samples, the Random Forest generates k bootstrap samples D_1, D_2, \dots, D_k by sampling with replacement from D . Each subset D_j is used to independently train one decision tree T_j :

$$D_j = \text{Bootstrap}(D, n) \quad (1)$$

Step 2: Feature Subset Selection

At each node in tree T_j , a random subset of m features is selected where $m < d$, decorrelating individual trees and preventing any single dominant feature from controlling splits across the entire ensemble.

$$m = \sqrt{d} \quad (2)$$

Step 3: Node Splitting via Gini Impurity

Each split is determined by minimizing the Gini impurity G across the selected feature subset. For a node containing samples from C classes:

$$G = 1 - \sum p_o^2 \quad (3)$$

where p_o is the proportion of samples belonging to class c at that node. The feature and threshold producing the lowest weighted Gini impurity across child nodes are selected for the split.

Step 5: Tree Construction

Each tree T_j grows recursively by applying Steps 2 and 3 until either a minimum node size is reached or all

samples at a node share the same class label. The output of each tree for input x is:

$$T_j(x) \in \{0, 1\} \quad (4)$$

Step 6: Ensemble Aggregation via Majority Voting

Predictions from all k trees are combined through majority voting. The final predicted label for input x is:

$$\hat{y} = \text{mode}\{T_1(x), T_2(x), \dots, T_k(x)\} \quad (5)$$

This aggregation lowers variance relative to any individual tree while keeping bias in check, giving Random Forest stronger generalization than standalone decision trees on unseen samples.

Step 7: Output and Confidence Scoring

The final output is the predicted class label \hat{y} alongside a confidence score derived from the fraction of trees voting for that class:

$$\text{Confidence}(x) = (1/k) \sum \mathbb{1}[T_j(x) = \hat{y}] \quad (6)$$

This confidence score is returned alongside the prediction label within the deployed Arcus application, giving users a quantitative measure of model certainty tied directly to their individual screening result. The model was instantiated with a fixed random seed of 42 and default Scikit-learn hyperparameters. A separate **GridSearchCV** was applied exclusively to the KNN classifier to identify the optimal value of k over the search space {3, 5, 7, 9}, yielding k = 7. The trained Random Forest model was serialized using **Joblib** for deployment within the Arcus backend.

E. Dataset Description

The dataset used in this study is the Autism Screening Adult dataset from the UCI Machine Learning Repository, consisting of 704 adult records collected through an online ASD screening survey. Each record contains responses to the ten AQ-10 behavioral items, A1_Score through A10_Score, encoded as binary values, alongside demographic attributes including age, gender, ethnicity, country of residence, family history of ASD, jaundice at birth, prior app usage, and the relationship of the respondent to the individual being screened. The target variable is a binary Class/ASD label. The dataset reflects a real-world class imbalance, with approximately 73.15% NO and 26.85% YES labels, which informed the decision to use F1-score and cross-validation mean as primary evaluation metrics alongside raw accuracy

Fig 2. Example samples of the dataset

F. Classification Metrics

Accuracy:

Accuracy measures the proportion of correctly classified

instances across both classes:

$$\text{Accuracy} = \frac{\text{Correct Prediction}}{\text{Total Predictions}} \quad (7)$$

Precision:

Precision measures the fraction of positive predictions that were actually correct, penalizing false positives:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (8)$$

Recall:

Recall measures the fraction of actual positive cases that were correctly identified, penalizing false negatives:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (7)$$

F1 Score:

F1-Score is the harmonic mean of precision and recall, providing a balanced metric particularly useful when class distributions are uneven:

$$F1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (8)$$

IV. RESULT ANALYSIS

TABLE II. PERFORMANCE COMPARISON OF CLASSIFICATION MODELS

MODEL	ACCURACY	PRECISION	RECALL	F1-SCORE	CV MEAN
Decision Tree	0.8794	0.91	0.88	0.88	0.9034
Random Forest	0.9929	0.99	0.99	0.99	0.9588
KNN	0.9007	0.90	0.90	0.90	0.8608
Naïve Bayes	0.3121	0.69	0.31	0.23	0.3494
Gradient Boosting	0.9574	0.96	0.96	0.96	0.9588
XGBoost	0.9787	0.98	0.98	0.98	0.9673

Six supervised classification algorithms were evaluated on the UCI Autism Screening Adult dataset under identical preprocessing and evaluation conditions. Performance was measured across accuracy, precision, recall, and F1-score on the held-out test set of 141 samples, alongside 5-fold cross-validation mean to assess generalization.

Decision Tree produced an accuracy of 87.94%, with the confusion matrix showing 90 true negatives, 34 true positives, 15 false positives, and 2 false negatives. While it handled the majority class reasonably well, the relatively high false positive count of 15 indicates a tendency to misclassify non-ASD individuals as ASD-positive, which would be a concern in a clinical screening context.

K-Nearest Neighbors, tuned to k = 7 via GridSearchCV, reached 90.07% accuracy. The confusion matrix recorded 99 true negatives, 28 true positives, 6 false positives, and 8 false negatives. The false negative count of 8, meaning 8 actual ASD-positive cases were missed, is worth noting; as missed positives carry a higher practical cost in screening applications than false positives.

Naïve Bayes performed substantially below all other classifiers, achieving only 31.21% accuracy. The confusion matrix revealed an extreme bias toward predicting the positive class, with 95 false positives out of 105 actual negatives. This collapse in performance is consistent with

the known sensitivity of Gaussian Naïve Bayes to correlated features, a condition that clearly holds on this dataset given the inter-related behavioral indicators.

Gradient Boosting achieved 95.74% accuracy with a confusion matrix of 99 true negatives, 36 true positives, 6 false positives, and 0 false negatives. The zero false negatives is a notable result, every actual ASD-positive case in the test set was correctly identified. However, 6 non-ASD individuals were incorrectly flagged, and the cross-validation mean of 95.88% was effectively identical to that of Random Forest, meaning it did not generalize better despite its sequential learning approach.

XGBoost reached 97.87% accuracy with 102 true negatives, 36 true positives, 3 false positives, and 0 false negatives. Like Gradient Boosting, it produced zero false negatives. Its cross-validation mean of 96.73% was the highest among all models, indicating strong generalization. However, it still fell short of Random Forest on test accuracy and F1-score.

Random Forest delivered the strongest overall performance across every metric. The confusion matrix confirmed 105 true negatives, 35 true positives, 0 false positives, and only 1 false negative across all 141 test samples, a single misclassification in the entire test set. It achieved 99.29% accuracy, a weighted F1-score of 0.99, and a cross-validation mean of 95.88%. The combination of near-perfect test accuracy, zero false positives, and strong cross-validation performance confirmed that the model was genuinely robust rather than simply overfitting to the test split. These results justified its selection as the deployment model for the Arcus application.

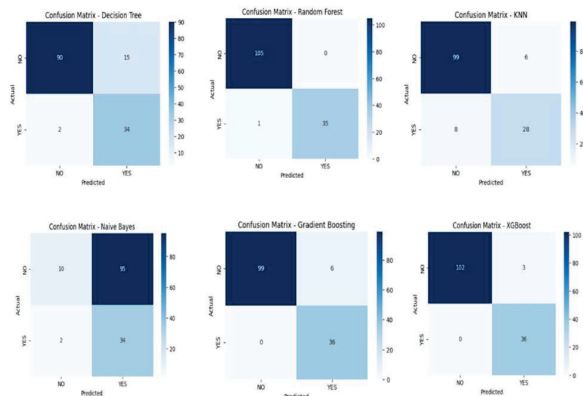


Fig 5. Confusion Matrices for All Six Classifiers

V. DISCUSSION

The results of this study point to a few things worth unpacking beyond the numbers themselves..

Random Forest's dominance across all three evaluation metrics was not entirely surprising given the nature of the dataset, but the margin was larger than expected. What stood out was not just the 99.29% accuracy but the zero false positive count, no non-ASD individual was incorrectly flagged across the entire test set. In a screening context, this matters more than raw accuracy. A tool that over-flags healthy individuals erodes trust quickly, particularly in a health-adjacent domain where a false positive can cause unnecessary anxiety. The single misclassification was a

false negative, one ASD-positive individual predicted as negative, which while imperfect, is a far more forgivable error in a preliminary screening tool that explicitly positions itself as a first filter rather than a diagnosis.

Naïve Bayes was a clear outlier. Its 31.21% accuracy traces directly to its core assumption, conditional feature independence, which breaks down on this dataset where the ten AQ-10 behavioral items are inherently correlated. A person who scores positively on one behavioral indicator tends to score positively on related ones. That correlation structure is precisely what Naïve Bayes cannot model, and the confusion matrix showed the consequences: 95 out of 105 negative cases were incorrectly classified as positive. This result serves as a useful reminder that model selection should always be validated empirically on each specific dataset rather than assumed from general reputation.

Gradient Boosting and XGBoost both produced competitive results and both achieved zero false negatives, meaning neither missed a single ASD-positive case. The tradeoff was a small number of false positives in each case. Between the two, XGBoost's higher cross-validation mean of 96.73% suggests slightly better generalization than Gradient Boosting's 95.88%. However, Random Forest outperformed both on test accuracy and matched Gradient Boosting on cross-validation mean, making it the strongest choice across the board.

The web application component of this work, Arcus, addressed something that the reviewed literature largely left unresolved: translating a trained model into something a non-technical person can actually use. The three-sprint agile delivery ensured that the system was built incrementally around real user needs rather than retrofitted with a frontend after the model work was complete. Every one of the ten committed user stories was delivered within its planned sprint, with no descoped features or carryovers. Security was treated as a first-class requirement from Sprint 2 onward, with JWT authentication and bcrypt password hashing in place before the system began handling any real screening data.

VI. LIMITATIONS

Several limitations of the current system are worth acknowledging.

The dataset used for training contains 704 records, which is relatively modest for a classification task with this many categorical features post-encoding. While the model performed strongly on the test split and across cross-validation folds, its generalization to genuinely diverse populations, across ethnicities, age ranges, and cultural contexts, cannot be confirmed without evaluation on larger, more representative datasets. Kohli et al. [4] raised similar concerns about cultural bias in screening instruments, and those concerns apply here.

The AQ-10 instrument itself is a self-report tool, meaning the quality of the prediction depends entirely on the honesty and self-awareness of the person completing the questionnaire. Individuals who lack insight into their own behavioral patterns, which is not uncommon in the ASD population, may provide responses that do not accurately reflect their actual traits, limiting the reliability of the prediction regardless of model accuracy.

The system currently supports adult screening only. The

dataset is drawn entirely from adult records, and the questionnaire items are framed for adult self-assessment. Extending the tool to pediatric screening would require a separate dataset, age-appropriate questionnaire items, and a different respondent model where parents or caregivers complete the form on behalf of the child.

The trained model was serialized and loaded into the Node.js backend via a Python inference service. While functional, this cross-language integration introduces a dependency that adds deployment complexity. A production-grade system would benefit from a more tightly integrated inference pipeline.

Finally, the application has not undergone formal clinical validation. Its predictions are grounded in a well-established screening instrument and a high-performing classifier, but no study has been conducted to compare its outputs against formal clinical diagnoses on a held-out population. Until such validation is done, the tool should be positioned strictly as a preliminary screener rather than a diagnostic aid.

VII. CONCLUSION

This paper presented an end-to-end ASD screening system built around two tightly coupled components, a comparative machine learning study and a full-stack web application named Arcus. Six classifiers were trained and evaluated on the UCI Autism Screening Adult dataset under consistent preprocessing and cross-validation conditions. Random Forest emerged as the strongest performer with 99.29% accuracy, an F1-score of 0.986, and a 5-fold cross-validation mean of 95.88%, producing only a single misclassification across 141 test samples. The trained model was deployed within Arcus, a React, Node.js, and PostgreSQL application that guides users through the AQ-10 questionnaire, returns a real-time prediction with confidence scoring, provides plain-language result explanations and clinical recommendations, and maintains a longitudinal screening history, all secured through JWT authentication and bcrypt password hashing.

The development followed an agile methodology across three sprints, with all ten committed user stories delivered within their planned sprint cycles. What this work demonstrates, beyond the model performance figures, is that the gap between a well-performing classifier and a tool that genuinely serves non-technical users is a real engineering and design challenge, one that most prior work in this space has not addressed. Arcus bridges that gap in a practical way, and the results suggest that ensemble-based ASD screening, when paired with a thoughtfully designed clinical interface, can serve as a meaningful and accessible first step toward early identification for individuals who might otherwise have no structured starting point.

Future work will focus on expanding the training dataset to improve demographic diversity, extending support to pediatric screening cohorts, integrating with electronic health record systems for direct clinical handoff, and pursuing formal clinical validation to establish the tool's reliability relative to professional diagnostic outcomes.

REFERENCES

[1]. L. Mertz, "Using AI and ML to predict autism spectrum disorder," *IEEE Pulse*, vol. 15, no. 5, pp. 11–15, Sep./Oct. 2024, doi: 10.1109/MPULS.2024.3443489.

[2]. M. M. V. Pinto, E. A. L. S. Arisawa, L. J. Raniero, and T. Bhattacharjee, "Saliva FTIR spectra and machine learning for autism spectrum disorder diagnosis—preliminary study," *IEEE Photon. J.*, vol. 17, no. 3, Jun. 2025, Art. no. 8500504, doi: 10.1109/JPHOT.2025.3561020.

[3]. S. M. M. Hasan, M. P. Uddin, M. A. Mamun, M. I. Sharif, A. Ulhaq, and G. Krishnamoorthy, "A machine learning framework for early-stage detection of autism spectrum disorders," *IEEE Access*, vol. 11, pp. 15038–15057, 2023, doi: 10.1109/ACCESS.2022.3232490.

[4]. M. Kohli, A. K. Kar, and S. Sinha, "The role of intelligent technologies in early detection of autism spectrum disorder (ASD): A scoping review," *IEEE Access*, vol. 10, pp. 104887–104906, 2022, doi: 10.1109/ACCESS.2022.3208587.

[5]. V. Yaneva, L. A. Ha, S. Eraslan, Y. Yesilada, and R. Mitkov, "Detecting high-functioning autism in adults using eye tracking and machine learning," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 6, pp. 1254–1261, Jun. 2020, doi: 10.1109/TNSRE.2020.2991675.

[6]. D. P. Kavadi, V. R. R. Chirra, P. R. Kumar, S. B. Veeram, S. Yeruva, and L. K. Pappala, "A hybrid machine learning model for accurate autism diagnosis," *IEEE Access*, vol. 12, pp. 194911–194930, 2024, doi: 10.1109/ACCESS.2024.3520009.

[7]. R. S. Parpinelli, G. L. Balatka, M. H. Toviansky, M. F. Gomes, T. M. B. nches, H. S. Alves, M. A. Correa, and I. A. Costa, "A data-driven approach for autism spectrum disorder screening," *IEEE Access*, vol. 14, pp. 4864–4880, 2026, doi: 10.1109/ACCESS.2026.3650856.

[8]. S. Saranya and R. Menaka, "A quantum-based machine learning approach for autism detection using common spatial patterns of EEG signals," *IEEE Access*, vol. 13, pp. 15739–15755, 2025, doi: 10.1109/ACCESS.2025.3531979.

[9]. M. Briguglio, L. Turriziani, A. Currò, A. Gagliano, G. Di Rosa, D. Caccamo, A. Tonacci, and S. Gangemi, "A machine learning approach to the diagnosis of autism spectrum disorder and multi-systemic developmental disorder based on retrospective data and ADOS-2 score," *Brain Sci.*, vol. 13, no. 6, p. 883, 2023, doi: 10.3390/brainsci13060883.

[10]. K. Vakadkar, D. Purkayastha, and D. Krishnan, "Detection of autism spectrum disorder in children using machine learning techniques," *SN Comput. Sci.*, vol. 2, no. 5, p. 386, 2021, doi: 10.1007/s42979-021-00776-5.

[11]. M. D. Hossain, M. A. Kabir, A. Anwar, and M. Z. Islam, "Detecting autism spectrum disorder using machine learning techniques: An experimental analysis on toddler, child, adolescent and adult datasets," *Health Inf. Sci. Syst.*, vol. 9, no. 1, p. 17, 2021, doi: 10.1007/s13755-021-00145-9.

[12]. D. Bone, M. S. Goodwin, M. P. Black, C.-C. Lee, K. Audhkhshi, and S. Narayanan, "Applying machine learning to facilitate autism diagnostics: Pitfalls and promises," *J. Autism Dev. Disord.*, vol. 45, no. 5, pp. 1121–1136, 2015, doi: 10.1007/s10803-014-2268-6.

[13]. F. Thabtah and D. Peebles, "A new machine learning model based on induction of rules for autism detection," *Health Inform. J.*, vol. 26, no. 1, pp. 264–286, 2020, doi: 10.1177/1460458218824711.

[14]. K. K. Hyde, M. N. Novack, N. LaHaye, C. Parlett-Pelleriti, R. Anden, D. Dixon, and E. Linstead, "Applications of supervised machine learning in autism spectrum disorder research: A review," *Rev. J. Autism Dev. Disord.*, vol. 6, no. 2, pp. 128–146, 2019, doi: 10.1007/s40489-019-00158-x.

[15]. D. P. Wall, J. Kosmicki, T. F. DeLuca, E. Harstad, and V. A. Fusaro, "Use of machine learning to shorten observation-based screening and diagnosis of autism," *Transl. Psychiatry*, vol. 2, p. e100, Apr. 2012, doi: 10.1038/tp.2012.10.