Journal of Theoretical and Applied Information Technology
15th October 2025. Vol.103. No.19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ADAPTIVE DEEP LEARNING FRAMEWORK USING BAYESIAN OPTIMIZATION FOR AUTISM SPECTRUM DISORDER PREDICTION FROM SCREENING DATA

B. DEEPA [1], Dr.K.S. JEEN MARSELINE [2]

[1]Assistant Professor, Bharathiar University, Department of Information Technology and Cognitive Systems, India

[2] Associate Professor. Bharathiar University, Department of Information Technology and Cognitive Systems, India

E-mail: [1]deepa1979b@gmail.com, [2]jeenmarselineks@skasc.ac.in

ABSTRACT

Autism Spectrum Disorder (ASD) is a complex neurodevelopmental condition marked by persistent challenges in communication, behaviour regulation, and social interaction. The heterogeneity of symptoms across individuals and age groups complicates early detection, as behavioural traits often overlap with other conditions or remain masked until later developmental stages. Traditional diagnostic methods are usually time-intensive, subjective, and rely on specialist interpretation, leading to delayed or inconsistent identification. Screening data offers a scalable, cost-effective, and non-invasive alternative for early ASD prediction, capturing observable traits through structured behavioural questionnaires. To overcome diagnostic inconsistencies and optimize model performance, this research proposes a Bayesian Optimization based Long Short-Term Memory (BO-LSTM) framework that adaptively learns temporal dependencies in screening responses while automatically tuning its parameters using a probabilistic surrogate model. The model was evaluated using the Autism Screening Dataset, comprising 6075 records and 20 structured attributes, sourced from a mobile-based application developed by Dr. Fadi Fayez. The dataset includes behavioural inputs from toddlers, children, adolescents, and adults, with questionnaires tailored to each age group. BO-LSTM achieved a classification accuracy of 74.375%, along with notable gains in sensitivity, specificity, and interpretability. These results demonstrate the framework's effectiveness in processing sequential screening data for timely and reliable ASD prediction across diverse age groups.

Keywords: *Autism Spectrum Disorder, Prediction, Screening Data, Deep Learning, Long Short-Term Memory, Bayesian Optimization*

1. INTRODUCTION

Autism is a lifelong neurodevelopmental condition marked by persistent challenges in communication, behaviour regulation, and social engagement. Its clinical presentation spans a spectrum, with individuals showing varying degrees of language delay, restricted interests, and sensory sensitivities [1]. Subtypes include conditions such as Asperger's syndrome and atypical autism, each reflecting different levels of functional ability and behavioural rigidity. Though grouped under the same spectrum, these subtypes exhibit non-uniform traits that resist generalization [2], [3]. This variability makes standardized diagnosis and prediction difficult, especially when behavioural expressions evolve across contexts. The distinction between mild and severe traits is not always detectable through single-session clinical assessments. Accurate identification of these subtypes is clinically significant and essential for tailoring early intervention strategies that align with individual support needs [4].

Autism Spectrum Disorder (ASD) affects individuals across all age groups, but the expression of symptoms and diagnostic clarity vary significantly with age [5], [6]. In infants and toddlers, delays in language acquisition, minimal social initiation, or atypical gaze patterns may be early indicators, yet these are often misinterpreted as personality differences or delayed development. Adolescents may present more subtle manifestations, including social withdrawal or obsessive focus, which may be masked or overlooked entirely [7]. Adults frequently remain undiagnosed due to learned compensatory behaviours or limited access to retrospective developmental evaluations. The symptoms overlap with other conditions like ADHD or anxiety, which complicates accurate detection. Gender-based diagnostic gaps and cultural differences in behaviour interpretation amplify these challenges [8]. The dynamic nature of symptom expression across time

introduces inconsistencies in clinical judgments, making it essential to explore models that adaptively capture behavioural patterns and predict ASD in temporally diverse populations [9].

ASD research commonly leverages two forms of data: imaging datasets such as fMRI or EEG and non-imaging, screening-based datasets collected through structured questionnaires or rating scales [10]. Imaging-based approaches offer neurophysiological insights but often require high-cost equipment and skilled operators, and they are less feasible for large-scale or early-stage screening. Screening datasets, in contrast, are low-cost, widely accessible, and capture observable traits grounded in lived behaviour [11]. These datasets mirror real-world symptomatology, enabling scalable, population-level ASD monitoring without specialized infrastructure [12]. This research focuses exclusively on screening datasets to ensure broad applicability and practical relevance, especially in resource-limited settings. Screening responses often carry temporal dependencies, and symptoms may evolve across repeated assessments or structured response sequences [13]. Unlike static neuroimages, these sequences demand models that learn from transitions, not isolated snapshots. The screening data choice supports ethical deployment and dynamic modelling of ASD behaviour, which is often shaped by time, context, and developmental stage.

Deep learning has emerged as a powerful approach in ASD prediction due to its capacity to capture complex, nonlinear relationships within behavioural data. Unlike traditional statistical methods, deep models can learn from raw, structured inputs such as screening questionnaires without requiring manual feature engineering [14]. Recurrent architectures have shown strength in modelling sequential dependencies, common in symptom progression or temporally structured assessment responses. These models help uncover latent behavioural trends that are not immediately apparent through static analysis. Probabilistic methods enhance this predictive capacity by incorporating uncertainty estimation into learning [15]. Techniques such as Gaussian Processes and Bayesian Optimization contribute by offering guided exploration of model parameters, reducing overfitting, and improving generalization. Their integration within behavioural health modelling provides a structured mechanism to handle variability in human-assessed data, making them suitable for sensitive domains like early ASD identification from screening evaluations [16], [17].

Bio-inspired optimization algorithms take inspiration from the adaptive behaviours, cooperative strategies, and survival mechanisms found in nature [18]-[27]. These methods are designed to efficiently explore and exploit complex search spaces, often achieving superior performance in solving high-dimensional, nonlinear, and multimodal optimization problems. Their stochastic and adaptive nature allows them to escape local optima, balance exploration with exploitation, and maintain robustness under uncertain or dynamic conditions [28]-[40]. Such characteristics make bio-inspired optimization a versatile tool that can be applied across various computational tasks, including feature selection, parameter tuning, and model enhancement, where traditional optimization methods may struggle [41]-[57].

1.1. Problem Statement

ASD diagnosis remains a significant challenge due to the spectrum's inherent heterogeneity, subtle behavioural markers, and overlapping traits with other neurodevelopmental conditions. Traditional assessments often rely on subjective interpretation of structured screening responses, leading to diagnostic delays and variability across age, gender, and cultural groups. Although non-imaging screening datasets offer scalable and accessible data sources, current computational models underutilize their temporal and decision-structured nature. Deep learning approaches, particularly recurrent architectures, show potential yet are constrained by sensitivity to hyperparameter configurations, lack of robustness, and poor generalization when applied to low-dimensional, sequential inputs. These models frequently require exhaustive manual tuning and fail to embed uncertainty estimation, reducing both efficiency and interpretability. Existing limitations restrict real-world deployment in early screening environments, particularly in low-resource or non-specialist settings. This exposes a critical gap in designing models that can adaptively, reliably, and scalably process behavioural screening data for accurate and timely ASD identification.

1.2. Motivation

Early prediction of ASD is critical for enabling timely interventions that support communication, cognitive development, and adaptive functioning. Missed or delayed diagnoses often result in long-term challenges for individuals and families, especially in communities with limited access to specialized assessment services. Structured screening tools provide a practical alternative, but existing models often underutilize their sequential

nature and behavioural complexity. Most predictive systems rely on static features and require extensive manual tuning, which hinders efficiency and scalability. There is a growing need for models that can intelligently process temporal screening data and adapt configurations without human intervention. Bridging this gap is vital to ensure consistent, interpretable, and accessible ASD detection across real-world settings, supporting early intervention pathways that are both equitable and actionable above the sub section while no space should be given below the heading and text

1.3. Objective

This research aims to design and validate an adaptive deep-learning framework for the early prediction of ASD using sequential responses from non-imaging screening assessments. The proposed methodology, Bayesian Optimization based Long Short-Term Memory (BO-LSTM), is developed to address critical limitations in existing models that fail to capture temporal dependencies and require extensive manual hyperparameter tuning. By integrating a probabilistic surrogate model and acquisition-guided sampling, BO-LSTM automatically identifies optimal configuration paths, improving training efficiency and model generalization. The framework is built to process real-world behavioural screening sequences with minimal overfitting, and its predictive performance is evaluated using classification accuracy as the primary metric. Interpretability is embedded through structured output mappings to ensure model transparency and clinical relevance. This research proposes a scalable, data-efficient, and reliable prediction model for early ASD identification across diverse and resource-constrained settings.

2. LITERATURE REVIEW

Imperialistic Competitive Feature Selector" [58] applies the Imperialistic Competitive Algorithm (ICA) for feature selection, simulating socio-political competition. Each country in the search space represents a candidate feature subset, and mighty empires assimilate weaker ones based on classification performance. "Relational Graph Attention Network" [59] models ASD classification using graph attention networks, where each node represents a subject and edges encode similarity based on phenotype or fMRI traits. The model learns attention scores that determine the influence of neighbouring nodes, with different attention weights applied depending on edge type. Multiple graph variants (phenotype-only, fMRI-only, combined) are constructed to evaluate relational strength. "Atypical Salient Region Enhancer (ASRE)" [60] uses an encoder-decoder architecture with intermediate modules to refine visual saliency detection for ASD individuals. Architecture handles abnormal attention distribution specific to ASD by adjusting feature fusion at each decoding layer, ensuring the final map reflects ASD-specific visual tendencies. All enhancements operate in a convolutional setting without recurrence or graph structures.

"Optimizer Ensemble Convolution Network" [61] constructs several CNN models with identical architectures but different optimization algorithms like Adam and Nadam. Each CNN is trained on structural MRI data, where on-the-fly augmentation applies spatial and intensity transformations in real-time. This strategy introduces diversity in learned weights. "Capsule Dense Network Reinforcer" [62] combines feature extraction with behavioural recommendation. Input features are first optimized using Cosmo Nest, a metaheuristic combining African Vulture and Butterfly behaviour to identify informative attributes. These are passed into Capsule Dense Net++, which uses capsule routing to preserve spatial hierarchies, and Dense Net layers to promote feature reuse. Classification identifies ASD status from screening data. "Federated Convolutional LSTM Network" [63] builds a decentralized model where each local node uses CNN to extract spatial features and LSTM to model behaviour sequences from screening data. These regional models train independently and transmit encrypted weights, not raw data, to a central server. The server aggregates them using federated averaging.

"Adaptive Fuzzy Reasoning Network" [64] applies a Takagi–Sugeno–Kang fuzzy inference system combined with contrastive domain adaptation for rs-fMRI-based ASD classification. Features are fuzzified using Gaussian membership functions, and fuzzy rules map input conditions to ASD or control labels. A domain adaptation module aligns cross-site feature distributions using contrastive learning. "Support Vector EEG Classifier" [65] processes task-evoked EEG data to differentiate low- and high-functioning autism. Signals are filtered and augmented using Gaussian bootstrapping. Band-specific features like absolute and relative power are computed from the delta to gamma bands. Derived features include theta-alpha and theta-beta ratios, reflecting cognitive workload. "Supervised Connectivity Model Survey" [66] reviews machine learning models trained on functional brain connectivity matrices derived from fMRI. Atlas-

based brain segmentation defines nodes and the correlation between regions forms features. These matrices are flattened into vectors input to models like SVM, decision trees, or ensemble learners. Recursive feature elimination and PCA are used for dimensionality reduction

"Dual Transformer Self Learner" [67] models repetitive behaviour detection using pose-estimated video frames. A dual-branch transformer processes spatial key points and temporal transitions using self-attention. Self-supervised proxy tasks such as frame order prediction, spatial jigsaw, and motion reconstruction train the network without manual labels. "Structural Equation Burnout Model" [68] builds a statistical framework using Partial Least Squares Structural Equation Modelling (PLS-SEM) to explore escapism in autistic gamers. Inputs include psychometric scales for autistic burnout, and gaming motivations. Latent variables such as self-suppression, self-expansion, and escapism are derived from observable questionnaire items. "Spatio Temporal Learning Network" [69] processes fMRI data through parallel branches for spatial and temporal feature learning. The spatial path uses CNN layers to encode region-specific activity, while the temporal path applies recurrent layers to capture dynamic fluctuations. Attention mechanisms enhance signal importance in both paths. A feature-sharing block transfers functional patterns between streams, and fused embeddings are passed to a classification head. Multi-task loss guides the training across spatial and temporal dimensions, aligning learned features with class labels

"Support Vector Machines (SVM)" [70] presents a classifier trained on questionnaire-derived screening data, stratified across toddler, child, and adult age groups. The process begins with feature selection using correlation metrics, isolating high-relevance behavioural indicators. These features are mapped into a kernel space where SVM identifies a hyperplane that maximizes the separation between ASD and non-ASD responses. Margin constraints and support vectors are adjusted per age group to accommodate developmental variation in symptom expression. "Big Data and Machine Learning-based Medical Data Classification (BDML-MDCASD)" [71] presents a hybrid architecture that begins with ISSA-FS for pruning irrelevant behavioural features. Each dataset—child, adolescent, and adult—is separately filtered using this swarm-inspired selection process. An Autoencoder encodes selected features into compressed latent vectors, which are then classified using a BOA-guided decision layer. The process is distributed across computing nodes using MapReduce to manage scale and ensure uniform processing.

2.1. Comparative Insights and Significance of Improvement

The reviewed approaches demonstrate valuable contributions to ASD prediction; however, many rely on neuroimaging modalities that are costly, resource-intensive, and impractical for large-scale screening. Non-imaging methods, while more accessible, often struggle with instability when confronted with noisy, incomplete, or imbalanced screening datasets. Several state-of-the-art classifiers lack mechanisms to adapt to behavioural drift across age groups or to control overfitting under limited data diversity. Others provide high accuracy on constrained datasets but show reduced generalisation across demographic or cultural variations. The proposed Lagrangian-optimised reinforcement learning framework directly addresses these gaps by embedding stability constraints, bias suppression, and entropy-based exploration into the learning process. This enables the model to maintain robust performance under challenging screening conditions where prior methods degrade. Comparative results confirm consistent gains in accuracy, balanced sensitivity and specificity, and resilience to data variability, highlighting a substantial advancement over existing techniques in both technical capability and real-world applicability for ASD screening.

3. PROPOSED METHODOLOGY

The proposed methodology introduces a Bayesian Optimization based Long Short-Term Memory (BO-LSTM) model designed for early prediction of Autism Spectrum Disorder using structured, non-imaging screening data. The architecture captures temporal patterns in sequential screening responses while automatically optimizing hyperparameters such as learning rate, dropout, unit size, and batch configuration. A Gaussian Process surrogate models validation loss and drives acquisition-guided sampling to identify high-performing configurations without exhaustive search. The framework balances predictive accuracy and model generalization across age-specific symptom profiles. This integrated approach improves stability, reduces manual intervention, and supports adaptive learning from behavioral data under real-world variability

3.1. Initialize Surrogate

The BO-LSTM framework depends on a surrogate model to efficiently approximate the validation loss

landscape during ASD classification using structured screening sequences. A Gaussian Process surrogate enables the system to capture nonlinear patterns and uncertainty while optimizing LSTM hyperparameters. Unlike exhaustive searches, this surrogate-based approach evaluates fewer configurations by predicting the performance landscape with limited real observations. This step becomes foundational in guiding successive Bayesian decisions in selecting optimal dropout rates, memory sizes, and learning rates. The surrogate directly interfaces with the ASD classification objective by favoring loss-minimizing LSTM configurations trained on temporal survey data

$$S: \theta \rightarrow \hat{y} \tag{1}$$

where S denotes the surrogate model that maps an LSTM hyperparameter vector θ to a predicted validation loss $\hat{y}$. This symbolic representation captures the surrogate's forecasting role during optimization.

Based on past observations, Gaussian Process regression constructs a posterior distribution over possible validation losses. Each LSTM configuration and its recorded loss update the GP model's belief about the function landscape. The posterior mean quantifies expected performance as a central indicator in acquisition evaluations. For ASD classification, it identifies configurations likely to generalize well on behavioural and diagnostic screening inputs. The posterior mean for a test point θ is analytically expressed using kernel relationships with prior evaluations.

$$\mu_t(\theta) = k_t(\theta)^{\intercal} A_t^{-1} y_t \tag{2}$$

where $\mu_t(\theta)$ represents the expected loss. The matrix $A_t = K_t + \sigma_n^2 I$ blends kernel-derived similarity with observation noise σ_n^2, while $K_t(\theta)$ encodes kernel similarity with existing configurations

Beyond estimating average performance, the surrogate model quantifies uncertainty at any candidate point. This variance measure helps balance exploration and exploitation during optimization. In ASD-related prediction tasks, certain hyperparameter regions may be sparsely explored; the surrogate variance ensures that such areas still have a chance to be sampled. The magnitude of uncertainty aids the acquisition function in preferring informative yet under-explored configurations that may capture subtle ASD-relevant patterns in input sequences.

$$\sigma_t^2(\theta) = k(\theta, \theta) - k_t(\theta)^{\intercal} A_t^{-1} k_t(\theta) \tag{3}$$

where $\sigma_t^2(\theta)$ incorporates both prior kernel values and the influence of previous observations, ensuring a calibrated estimate of model confidence.

The surrogate's predictive capability critically depends on how similarity is encoded between LSTM configurations. For ASD screening data, subtle shifts in dropout rates or hidden unit sizes can lead to significant accuracy changes. The Matern kernel captures such sensitivities while remaining flexible across configurations. This kernel quantifies the relation between two hyperparameter sets, forming the foundation for posterior mean and variance computations.

$$k(\theta_i, \theta_j) = \alpha^2 \left(1 + \frac{\sqrt{5} d_{ij}}{l} + \frac{5 d_{ij}^2}{3 l^2}\right) exp\left(-\frac{\sqrt{5} d_{ij}}{l}\right) \tag{4}$$

where d_{ij} is the Euclidean distance between configurations θ_i and θ_j,α defines the signal scale, and l controls the smoothness across hyperparameter transitions.

The surrogate model must continuously evolve as more LSTM configurations are evaluated. Once a new candidate configuration is trained and its validation loss recorded, this data is appended to the set of observations. The surrogate is then refitted to this augmented dataset. This update ensures that the GP posterior accurately reflects the current understanding of the performance surface. In ASD screening prediction, the updated surrogate improves decision quality by steering future evaluations toward more promising areas

$$D_{t+1} = D_t \cup \{(\theta_{t+1}, y_{t+1})\} \tag{5}$$

This update rule describes the incremental dataset D_{t+1} used to refit the GP. Here, θ_{t+1} is the newly sampled configuration, and y_{t+1} is its observed validation loss

3.2. Defining Search Space

The effectiveness of Bayesian Optimization in refining LSTM architecture relies critically on the boundedness and granularity of its search space. The search space defines the domain where the acquisition function proposes candidate configurations to train and evaluate. For ASD

classification tasks using structured screening datasets, the need for tightly controlled ranges stems from the discrete and low-dimensional nature of the input features. Optimal performance from the BO-LSTM arises when this space is neither overly broad nor overly restrictive, ensuring meaningful exploration while avoiding irrelevant or impractical regions.

$$H = \begin{Bmatrix} \eta \in [10^{-5}, 10^{-2}], \delta \in [0.1,0.5], \\ u \in [32,256], b \in [16,128] \end{Bmatrix} \quad (6)$$

where η denotes learning rate bounds, δ refers to dropout rate limits, u captures hidden unit choices, and b describes batch size intervals. These variables are the core LSTM hyperparameters most sensitive to classification variance across ASD profiles

Search space representation impacts the surrogate's modeling fidelity. For hyperparameters such as dropout and learning rate, a continuous representation allows finer resolution in predictive tuning. On the contrary, hidden unit sizes and batch sizes often benefit from discrete step-wise encoding, as underlying hardware optimizations prefer specific sizes. During BO, Gaussian Processes handle continuous dimensions natively, while discrete encodings are incorporated using indicator functions or integer mappings to maintain compatibility with probabilistic modeling

$$\theta = [log_{10}(\eta), \delta, log_2(u), log_2(b)] \quad (7)$$

where, θ transforms each hyperparameter into a scaled vector suitable for Gaussian Process regression. Using logarithmic terms stabilizes kernel evaluation by reducing extreme variance across numeric magnitudes.

Prior distributions are optionally embedded into the initial sampling mechanism to reinforce the Bayesian aspect of the optimization process. This step gives weight to empirically favorable hyperparameter regions, allowing the optimization to converge faster toward promising areas. In screening-based ASD datasets, prior belief can be derived from earlier LSTM trials or adjacent behavioural prediction models. These priors are often selected as log-uniform or beta distributions over the search space, enhancing the surrogate model's ability to distinguish between likely and unlikely configurations

$$P(\theta) = \prod_{i=1}^{4} p_i(\theta_i) \quad (8)$$

where $P(\theta)$ defines the joint prior distribution over the BO-LSTM hyperparameter vector. Each marginal prior $p_i(\theta_i)$ corresponds to one search space dimension, enabling the surrogate to integrate historical knowledge into its inference

The search space can be progressively refined during optimization by applying adaptive constraints. In BO-LSTM, once early iterations indicate certain regions consistently yield suboptimal loss, those regions are dynamically masked or penalized in the acquisition function. For example, dropout values close to zero may be consistently associated with overfitting on ASD datasets. Applying domain restrictions avoids such configurations in later stages, preserving optimization budget for meaningful exploration

$$H^* = \{\theta \in H: \mathbb{I}[L(\theta) < \lambda] = 1\} \quad (9)$$

where, H^* is a constrained space, and it filters configurations θ based on their associated loss $L(\theta)$, compared to a dynamic loss threshold λ. The indicator function $I[\cdot]$ activates only the regions deemed feasible by empirical observation

Gaussian Process surrogates used in BO require normalized input vectors to ensure stable kernel evaluations. Feature scaling standardizes each hyperparameter to a bounded interval, typically [0,1]. In ASD classification models, this transformation helps treat learning and dropout rates on an equal scale despite their native value. difference. Normalization also reduces numerical instability in matrix inversion processes during GP updates.

$$\theta' = \frac{\theta - \min(H)}{\max(H) - \min(H)} \quad (10)$$

where θ' is a normalized version of the raw hyperparameter vector θ, obtained through min-max scaling across each dimension. This vector becomes the final input to the surrogate model and acquisition function.

3.3. Initial Sample Points

The initial points serve as the foundation for the surrogate model in the Bayesian Optimization loop. Without reliable and well-distributed initial configurations, the GP surrogate lacks the empirical structure required to predict validation loss accurately for unseen LSTM hyperparameters. For ASD classification using non-imaging screening datasets, the diversity of initial LSTM settings becomes essential to capture the variable expressiveness of behavioural sequence patterns. These samples help calibrate the initial posterior

belief, enabling the optimizer to identify candidate models with strong generalization potential early

$$\Theta_0 = \{\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(k)}\} \quad (11)$$

where Θ_0 denotes the initial configuration pool, where each $\theta^{(i)}$ represents a unique LSTM hyperparameter vector sampled within the bounded space H established in Step 2.

Latin Hypercube Sampling (LHS) is widely recognized as a space-filling design strategy suitable for generating diverse hyperparameter configurations. In the BO-LSTM context, LHS ensures that each hyperparameter range is proportionally explored across the initial configurations. This is particularly crucial for models applied to ASD screening datasets, where minor dropout rates or memory cell size changes may lead to significant classification shifts. LHS avoids clustering in any region and supports uniform exploration for optimal initialization of the Gaussian Process surrogate

$$\theta_j^{(i)} = \min_j + \left(\frac{\pi_j^{(i)} + \epsilon}{k}\right)\left(\begin{matrix} max \\ j \end{matrix} - \begin{matrix} min \\ j \end{matrix}\right) \quad (12)$$

where $\theta_j^{(i)}$ is the value of the j-th hyperparameter in the i-th configuration. The term $\pi_j^{(i)}$ is a unique permutation of the integers 0 through $k-1$, ϵ is a random variable in [0,1), and min_j,max_j are the bounds of the j-th hyperparameter.

The number of initial samples directly correlates with the dimensionality of the hyperparameter search space. For Bayesian Optimization to produce accurate mean and variance predictions, the surrogate must first be conditioned on sufficient evidence. In ASD classification models involving four primary LSTM parameters, a minimum of ten diverse configurations is typically adequate to begin surrogate modeling. This number is often derived from heuristics balancing exploration with computational expense.

$$k = \lceil \beta \cdot D \cdot \log(D) \rceil \quad (13)$$

where, k is the total number of initial samples, D is the number of dimensions (hyperparameters), and β is a constant controlling sampling density. This expression aligns the sample size with the complexity of the optimization domain.

Each sampled configuration must be stored in a matrix format for integration into the surrogate training process. This matrix becomes the initial training input for the Gaussian Process, influencing its kernel matrix, posterior mean, and uncertainty predictions. The matrix formulation also standardizes input data for efficient batch validation of initial LSTM trials. Each row corresponds to a configuration, and each column denotes one hyperparameter dimension across the normalized search space.

$$\Theta_0 = \begin{bmatrix} \theta_1^{(1)}\theta_1^{(1)} & \dots & \theta_D^{(1)} \\ \theta_1^{(2)}\theta_1^{(2)} & \dots & \theta_D^{(2)} \\ \vdots & \vdots & \vdots \\ \theta_1^{(k)}\theta_2^{(k)} & \dots & \theta_D^{(k)} \end{bmatrix} \quad (14)$$

where Θ_0 is the complete set of sampled points, where each $\theta_j^{(i)}$ is already normalized using the min-max transformation defined in Step 2.

Each initial LSTM configuration must be mapped to a corresponding validation loss via empirical training and testing. For screening-based ASD classification, this requires fitting each model to the dataset and capturing how well it learns sequential behavioural cues. These results form the target for fitting the Gaussian Process surrogate in Step 4. Accurate performance mapping at this stage enables more informed predictions over unexplored regions in the search space.

$$Y_0 = \{y^{(1)}, y^{(2)}, \dots . y^{(k)}\}, \; y^{(i)} = L(\theta^{(i)}) \quad (15)$$

where Y_0 is a target vector that contains validation losses, where each $y^{(i)}$ is computed as the loss for the LSTM model trained using the configuration $\theta^{(i)}$. This outcome provides the first layer of supervision for the surrogate model.

The quality of initial sampling can be measured using a coverage metric that quantifies the dispersion of the sampled configurations across the entire search space. For ASD classification, achieving optimal dispersion ensures that different behavioural patterns and response gradients across the screening dataset are adequately represented. The minimal pairwise Euclidean distance across the configuration matrix is a widely used coverage metric.

$$C(\Theta_0) = \min_{i \neq j} \left\| \theta^{(i)} - \theta^{(j)} \right\|_2 \quad (16)$$

where $C(\Theta_0)$ computes the minimum distance between any two distinct configurations. A larger value indicates better coverage and less redundancy in the initial sample set

3.4. Train LSTM Samples

The Bayesian Optimization framework requires empirical observations of validation loss corresponding to sampled LSTM configurations to construct an accurate surrogate model. Each configuration derived in Step 3.3 must be deployed within a complete training pipeline to extract real-world performance values. In the context of ASD classification from screening datasets, these values measure how effectively a given LSTM configuration models sequential input derived from structured behavioural patterns. The training phase supplies the ground truth against which the surrogate calibrates its mean prediction and confidence region.

$$M_{\theta^{(i)}} \leftarrow TrainLSTM\left(x, y; \theta^{(i)}\right) \quad (17)$$

where $TrainLSTM$ maps the input-output pairs (x, y), sampled from the screening dataset, to a trained model $M_{\theta^{(i)}}$ using hyperparameter configuration $\theta^{(i)}$.

Every LSTM model must be optimized using a differentiable loss function that reflects classification correctness. For ASD diagnosis tasks, binary cross-entropy is frequently employed due to the binary nature of the classification target (i.e., ASD or non-ASD). The loss surface defined by this function determines the gradient flow during optimization, impacting model convergence and generalization capacity. The validation loss derived from this function directly becomes the scalar target fed into the Bayesian surrogate.

$$L\left(\theta^{(i)}\right) = -\frac{1}{N}\sum_{n=1}^{N}\left[\begin{matrix} y_n log(\hat{y}_n) + \\ (1-y_n) log(1-\hat{y}_n)\end{matrix}\right] \quad (18)$$

where $L\left(\theta^{(i)}\right)$ represents the average binary cross-entropy over the validation set. Each $\hat{y}_n$ is the predicted probability output of the LSTM model $M_{\theta^{(i)}}$, corresponding to the ground truth label y_n

Sequential behavioural screening data exhibit temporal patterns, often reflecting decision sequences, checklist responses, or time-tagged observations. The LSTM architecture processes such data by maintaining the memory of previous steps using gated cell states. For every configuration sampled, the model must learn a stable mapping from these sequences to class labels, capturing nuanced transitions present in the data. This dynamic is governed by the hidden states evolving through the time axis of input data.

$$h_t^{(i)} = LSTMCell\left(x_t, h_{t-1}^{(i)}, c_{t-1}^{(i)}; \theta^{(i)}\right) \quad (19)$$

where $h_t^{(i)}$ and $c_t^{(i)}$ denote the hidden and cell states of the LSTM at time t, generated using the configuration $\theta^{(i)}$ and the input sequence element x_t. This recursive behaviour forms the core computation of each model variant under evaluation.

Training must be executed with controlled dataset partitioning to ensure unbiased performance evaluation. A common approach involves stratifying the screening dataset into training and validation subsets. Once the LSTM completes training on the training portion, the validation subset is passed through the model to compute predictive probability and, ultimately, the loss score. This process is standardized across all configurations to maintain consistency in surrogate updates.

$$(x^{train}, y^{train}), (x^{val}, y^{val}) = Split(x, y; \alpha) \quad (20)$$

where, Split separates the input dataset into training and validation segments according to a stratification ratio α, preserving class distribution for ASD labels across both partitions.

The recorded validation losses must maintain numeric smoothness and avoid excessive outlier influence to ensure compatibility with the surrogate's assumptions. Smoothing techniques or averaging over multiple runs per configuration are adopted in ASD classification tasks where stochastic training effects can skew single-run results. This smoothing enhances the Gaussian Process regression's ability to fit a reliable mean function.

$$\bar{L}\left(\theta^{(i)}\right) = \frac{1}{R}\sum_{r=1}^{R} L^{(r)}\left(\theta^{(i)}\right) \quad (21)$$

where, $\bar{L}\left(\theta^{(i)}\right)$ is the average loss over R repeated training trials using configuration $\theta^{(i)}$. Each $L^{(r)}$ reflects the result of an independent training pass, controlling randomness in weight initialization or data batching.

Every computed validation loss must be stored alongside its corresponding configuration to populate the data set required for surrogate fitting and acquisition optimization. This record becomes the empirical backbone of the Bayesian Optimization routine, supplying both training data and posterior calibration points.

$$D_0 = \{(\theta^{(i)}, \bar{L}(\theta^{(i)}) | i = 1, \dots, k\ )\} \quad (22)$$

The dataset D_0 each LSTM hyperparameter vector and its smoothed validation loss, forming the observed pairs defining the surrogate model's initialization data.

3.5. Fit Gaussian Process

The Bayesian Optimization process relies on a surrogate model to approximate the loss surface of the LSTM classifier trained on ASD screening datasets. The Gaussian Process (GP) is the probabilistic surrogate, offering a flexible non-parametric method to interpolate observed configuration-performance pairs and forecast the response at unseen locations. The model incorporates both the mean prediction and a variance estimate, enabling confidence-aware acquisition decisions in subsequent steps. The previously computed dataset $D_0 = \left\{\left(\theta^{(i)}, \bar{L}(\theta^{(i)})\right)\right\}$ now becomes the core evidence to condition the GP's posterior distribution over the hyperparameter domain.

$$y_0 = \left[\bar{L}(\theta^{(1)}), \bar{L}\theta^{(2)}, \dots., \bar{L}(\theta^{(k)})\right]^T \quad (23)$$

where vector y_0 contains all recorded validation losses, serving as target outputs for the surrogate regression task. These values anchor the GP predictions in empirically observed behaviour.

Gaussian Process inference requires the computation of a covariance matrix that captures pairwise similarities between hyperparameter configurations. Each entry in this matrix represents the kernel-derived similarity between two LSTM parameter vectors, where the kernel reflects prior assumptions about function smoothness and relevance of dimension-wise changes. For ASD classification tasks, modeling sensitivity in dropout, learning rate, and memory depth is particularly important to shape predictive uncertainty across the feature space.

$$K_{0,ij} = k(\theta^{(i)}, \theta^{(j)}; \phi) \quad (24)$$

where, each element in $K_{0,ij}$ of the covariance matrix is computed using the kernel function $\kappa(\cdot,\cdot)$, parameterized by hyperparameters ϕ. These parameters include signal variance and length scale and are typically optimized during marginal likelihood estimation.

Once the kernel matrix is defined and observations are incorporated, the surrogate must be updated to predict expected validation losses for any new LSTM configuration. The posterior mean estimates the central tendency of loss outcomes for a given hyperparameter input. For BO-LSTM in ASD screening classification, this estimate directs the acquisition function toward promising configurations that minimize prediction error for unseen cases.

$$\hat{\mu}(\theta_*) = k_*^{\intercal}(K_0 + \sigma_n^2 I)^{-1} y_0 \quad (25)$$

where $\hat{\mu}(\theta_*)$ denotes the surrogate's predicted mean loss at a new configuration θ_*. The vector k_* contains kernel values between θ_* and each point in D_0, while σ_n^2 represents noise variance accounting for stochastic effects during LSTM training.

Predictive uncertainty is integral to Bayesian Optimization, as it helps prioritize regions with high information gain. BO-LSTM enables focused exploration of hyperparameter regions with high epistemic uncertainty—an essential trait for identifying robust configurations that perform reliably on sequential behavioural datasets. The variance term quantitatively reflects this uncertainty and allows the acquisition function to incorporate exploitation and exploration.

$$\hat{\sigma}^2(\theta_*) = k(\theta_*, \theta_*) - k_*^{\intercal}(K_0 + \sigma_n^2 I)^{-1} k_* \quad (26)$$

where $\hat{\sigma}^2(\theta_*)$ gives the predicted variance for configuration θ_*. This variance peaks in under-explored regions, ensuring that such areas are not ignored in future sampling rounds.

The Gaussian Process surrogate includes internal hyperparameters, such as kernel length scales and signal variance, that must be learned from the data. Optimizing these values improves model fidelity to the observed data while controlling overfitting. The marginal log-likelihood expresses the plausibility of observed losses under the GP model and serves as the objective for internal tuning. In BO-LSTM, this alignment ensures that the surrogate conforms to the empirical structure of loss behaviour over the ASD classification space.

$$logp(y_0|\Theta_0) = -\frac{1}{2} y_0^{\intercal}(K_0 + \sigma_n^2 I)^{-1} y_0 - \frac{1}{2} log|K_0 + \sigma_n^2 I| - \frac{k}{2} log 2\pi \quad (27)$$

This expression measures how well the GP model fits the current dataset. Optimization of this log-likelihood refines kernel hyperparameters, directly improving the accuracy of mean and variance predictions during acquisition.

After surrogate fitting, the predicted mean scores across candidate configurations are normalized to

enable fair comparison across BO iterations. Normalization maps the scores into a bounded scale, allowing acquisition metrics to operate stably across varying loss ranges. For ASD screening analysis, such normalization balances the influence of extreme outliers or noise spikes, ensuring continuity in optimization behaviour.

$$s(\theta_*) = \frac{\hat{\mu}(\theta_*) - min(\hat{\mu})}{max(\hat{\mu}) - min(\hat{\mu})} \quad (28)$$

where, $s(\theta_*)$ represents the scaled loss prediction between 0 and 1, facilitating the acquisition strategy to interpret relative loss magnitudes rather than raw values.

3.6. COMPUTE ACQUISITION

The acquisition function is a probabilistic utility that guides the selection of the following configuration in Bayesian Optimization. It utilizes the Gaussian Process's predictive outputs, balancing exploiting low-loss regions and exploring uncertain configurations. In ASD classification using LSTM over structured screening data, the acquisition function ensures each subsequent training trial contributes meaningfully to discovering an optimal architecture. This controlled sampling reduces redundant evaluations and accelerates the discovery of generalizable models. The equation below defines the Expected Improvement (EI) acquisition function.

$$\alpha(\theta) = E[max(f^* - f(\theta), 0)] \quad (29)$$

where, f^* represents the best-observed validation loss, and $f(\theta)$ denotes the surrogate-predicted loss at configuration θ. The function measures the expected gain from evaluating θ compared to the current best.

The practical computation of EI requires converting the surrogate mean and variance predictions into a closed-form expression. This transformation facilitates efficient evaluation across the whole hyperparameter space. In the context of BO-LSTM, EI enables priority ranking of configurations for training, allowing the model to focus its computational budget on candidates with the highest potential for loss minimization in the ASD task.

$$EI(\theta) = (f^* - \mu(\theta)\Phi(Z)) + \sigma(\theta)\phi(Z) \quad (30)$$

where, EI using the Gaussian CDF $\Phi(Z)$ and PDF $\phi(Z)$. The variable Z represents the standardized improvement defined by the surrogate's posterior mean $\mu(\theta)$ and standard deviation $\sigma(\theta)$. The function quantifies how promising a configuration is relative to current knowledge.

The term Z used within the EI computation reflects the standardization of predicted loss difference. This adjustment ensures the acquisition function remains scale-invariant and maintains proportionality across diverse configurations. For BO-LSTM applied to ASD screening, such normalization is essential to compare LSTM setups fairly, especially across regions of differing dropouts or unit sizes.

$$Z = \frac{f^* - \mu(\theta)}{\sigma(\theta) + \epsilon} \quad (31)$$

where, Z is computed as the normalized improvement margin. The constant ϵ ensures numerical stability by avoiding division by zero, particularly in configurations with low predictive variance.

Alternative acquisition strategies, such as the Upper Confidence Bound (UCB), prioritize exploration explicitly. UCB adds a weighted uncertainty term to the predicted loss, promoting configurations with high epistemic uncertainty. This variant is advantageous during early optimization cycles when the surrogate lacks complete knowledge of the performance surface across the LSTM hyperparameter space.

$$UCB(\theta) = \mu(\theta) - k.\sigma(\theta) \quad (32)$$

where, κ is an exploration coefficient controlling the trade-off between mean prediction and uncertainty, a higher κ biases the sampling toward areas with greater variance, improving global coverage in the ASD classification task.

Another probabilistic acquisition function is the Probability of Improvement (PI), which focuses solely on the likelihood of achieving a better result than the current best. While not accounting for the magnitude of improvement, PI serves well when computational budgets are constrained, and simple decision rules are preferred. In BO-LSTM, PI may be applied during later stages to refine tuning in a localized region of the hyperparameter space.

$$PI(\theta) = \Phi\left(\frac{f^* - \mu(\theta)}{\sigma(\theta)}\right) \quad (33)$$

The PI acquisition function calculates the probability that configuration θ will improve upon the best-known result. It uses the CDF of the standard normal distribution over the standardized prediction gap.

3.7 Select Next Point

The acquisition function computed in the previous step provides a scalar utility value for each

configuration within the bounded hyperparameter search space. These values encapsulate a trade-off between exploiting known low-loss regions and exploring uncertain, potentially promising zones. The configuration associated with the highest acquisition score must be isolated to proceed with the subsequent model training trial in the BO-LSTM pipeline. This point represents the next LSTM candidate for empirical evaluation in the ASD classification process, ensuring that each decision is grounded in probabilistic guidance.

$$\theta^{next} = \arg\max_{H} \alpha(\theta) \quad (34)$$

where θ^{next} refers to the optimal candidate selected for evaluation, where $\alpha(\theta)$ is the acquisition score obtained from the surrogate. The maximization ensures that only the most information-rich configuration is forwarded to the training stage.

The acquisition function is generally non-convex across high-dimensional search spaces, particularly in LSTM tuning for complex screening data. To avoid local optima, a multi-start approach is implemented, where the maximization of the acquisition function begins from multiple random initializations. This improves the likelihood of converging to a globally optimal candidate. The strategy is crucial in BO-LSTM, where dropout rates and memory cell sizes can lead to intricate loss landscapes.

$$\theta^{start} = \left\{\theta_0^{(1)}, \theta_0^{(2)}, \ldots\ldots, \theta_0^{(m)}\right\} \quad (35)$$

where θ^{start} defines multiple random initialization points for optimizing $\alpha(\theta)$. Each of these seeds undergoes a local maximization procedure, improving robustness in the final selection.

Gradient-based methods can be used to improve the efficiency of acquisition maximization, especially under the Expected Improvement and UCB frameworks, which are differentiable. The optimal region in the hyperparameter space can be more quickly located by leveraging gradient ascent from each initialization point. The gradient at each point reflects the direction of the steepest ascent in acquisition value and is computed using the chain rule of the GP's mean and variance expressions.

$$\nabla_\theta \alpha(\theta) = \frac{\partial \alpha(\theta)}{\partial \mu(\theta)} \cdot \nabla_\theta \mu(\theta) + \frac{\partial \alpha(\theta)}{\partial \sigma(\theta)} \cdot \nabla_\theta \sigma(\theta) \quad (36)$$

where $\nabla_\theta \alpha(\theta)$ is composed of contributions from the surrogate model's mean and standard deviation. This gradient informs each optimization trajectory during selection.

The hyperparameter domain H often includes constraints, such as integer requirements for batch sizes or upper limits on dropout. These constraints must be enforced during acquisition maximization to ensure the following configuration remains valid. Constraint satisfaction can be integrated into the optimization process using penalty functions or projection operators that restrict updates to the feasible region.

$$\theta^{valid} = Project_{H_{valid}}(\theta^{next}) \quad (37)$$

where $Project_{H_{valid}}$ maps an unconstrained configuration back into the feasible domain H_{valid}, ensuring that the subsequent training trial does not violate structural or computational limits.

Configurations with extremely low predictive confidence are filtered out, even if they exhibit high acquisition scores. This filtering avoids evaluating hyperparameter combinations for which the GP surrogate has unreliable posterior estimates. The standard deviation from the GP model is used to enforce a minimum confidence threshold for selection.

$$\theta^{next} \in \{\theta | \sigma(\theta) < \tau\} \quad (38)$$

The confidence filter restricts candidate selection to those with predictive uncertainty $\sigma(\theta)$ below a threshold τ. This step enhances stability and reliability in hyperparameter exploration, especially in ASD screening datasets with sparse features.

A diversity penalty is introduced during acquisition maximization to avoid repetitive sampling in already-explored regions of the search space. This encourages exploration by penalizing candidates too close to previously evaluated configurations. Such a strategy is proper when the search space is dense and subtle variations in LSTM settings yield minimal new information.

$$\alpha'(\theta) = \alpha(\theta) - \lambda \cdot \min_i \left\|\theta - \theta^{(i)}\right\|_2 \quad (39)$$

The adjusted acquisition function $\alpha'(\theta)$ includes a penalty based on the Euclidean distance to the closest evaluated configuration. The penalty weight λ controls the trade-off between exploitation and diversity.

Acquisition scores are normalized into a probability distribution to accommodate probabilistic selection rather than greedy

maximization. This soft selection method avoids overcommitting to a single configuration and introduces stochasticity into the optimization process. This is especially beneficial in early iterations of BO-LSTM.

$$P(\theta) = \frac{exp(\alpha(\theta)/\gamma)}{\sum_{\theta' \in H} exp(\alpha(\theta')/\gamma)} \quad (40)$$

The probability $P(\theta)$ is derived using a softmax over scaled acquisition scores, where γ is a temperature parameter that controls the sharpness of the distribution. Lower γ values result in more deterministic selections.

3.8 Train with Suggestion

The configuration selected by maximizing the acquisition function in Step 7 must now be embedded into the LSTM training process. This configuration includes specific values for dropout rate, number of units, batch size, and learning rate, which were predicted to offer optimal classification performance for the ASD screening dataset. This configuration is instantiated into the LSTM architecture, initiating a fresh training cycle under the new hyperparameter regime. The training process aims to validate whether the theoretical advantage inferred by the surrogate and acquisition is supported by empirical accuracy.

$$M_{\theta^{next}} = Train(x^{train}, y^{train}, \theta^{next}) \quad (41)$$

This formulation describes the instantiation of the model $M_{\theta^{next}}$, trained using inputs x^{train}, corresponding targets y^{train}, and the selected configuration θ^{next}.

Once the model is instantiated, the temporal behavioural screening sequences are fed into the LSTM layer. The forward pass generates activations for each time step, maintaining hidden and cell states across the sequence. This mechanism enables the LSTM to internalize dependencies between sequential responses in the screening dataset, capturing the hidden progression of ASD-relevant traits. Each candidate configuration modifies the structure and behaviour of this temporal modeling, impacting learning dynamics and generalization. The below equation represents the LSTM hidden state update at time t.

$$h_t = f_h(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (42)$$

where h_t denotes the output, x_t is the input token, and W_{xh}, W_{hh}, b_h are the configuration-specific weight and bias terms. The activation function f_h is typically a nonlinearity such as $tanh$.

After processing the input sequence through the LSTM layers, the final hidden state is passed to a dense layer for classification. For binary ASD diagnosis, the SoftMax or sigmoid activation function converts the final state into a class probability. The effectiveness of this layer depends on the selected configuration's compatibility with the dataset's complexity and granularity.

$$\hat{y}_n = \sigma(W_o h_T + b_0) \quad (43)$$

where $\hat{y}_n$ is the predicted probability for the n-th instance, h_T is the final LSTM output, W_o and $b_0$represent weights and biases, and σ denotes the logistic sigmoid activation.

The learning process uses backpropagation through time (BPTT) and a configuration-specific optimizer to adapt model weights. Learning rate and batch size are critical to the pace and stability of convergence. Training continues for a fixed number of epochs or until early stopping criteria are met. The configuration applied directly determines the optimizer's response characteristics and gradient update behaviour. The gradient descent update rule in the below equation applies for each training iteration.

$$\theta_t = \theta_{t-1} - \eta \cdot \nabla_\theta L \quad (44)$$

where, η is the configuration-defined learning rate, θ_t is the model parameter at iteration $\nabla_\theta L$ is the gradient of the loss concerning that parameter.

Post-training, the model is evaluated on the validation subset to determine its generalization capability. The key metric computed is the validation loss, which forms the quantitative feedback for the Bayesian Optimization loop. This feedback enables surrogate recalibration in the next iteration and ensures that the acquisition-based decision is validated empirically in the ASD screening context. The loss function measures prediction error over N validation instances.

$$L_{val}(\theta^{next}) = \frac{1}{N}\sum_{n=1}^{N}\left[\begin{matrix} y_n log\hat{y}_n + \\ (1-y_n)log(1-\hat{y}_n)\end{matrix}\right] \quad (45)$$

where y_n is the actual label, $\hat{y}_n$ is the predicted probability output by the LSTM model trained with θ^{next}..

Each configuration may yield slightly different results due to stochasticity in weight initialization and mini-batch ordering. The training process is repeated multiple times for the same configuration to reduce variance in surrogate updates. The final loss used for surrogate fitting is the average across all repetitions, providing a stable representation of that configuration's utility in the optimization loop. The average validation loss is computed using the below equation over R independent training runs.

$$\bar{L}(\theta^{next}) = \frac{1}{R}\sum_{r=1}^{R} L_{val}^{(r)}(\theta^{next}) \quad (46)$$

where, each run r produces a validation score $L_{val}^{(r)}$, capturing a separate stochastic realization of training dynamics.

The newly observed performance of the suggested configuration must be appended to the surrogate's training dataset for posterior recalibration. This new pair augments the surrogate's information base and improves its ability to estimate future acquisition values. This incremental update enables BO-LSTM to refine its understanding of which hyperparameter regimes are most conducive to accurate ASD screening classification.

$$D_{t+1} = D_t \cup \{(\theta^{next}, \bar{L}(\theta^{next}))\} \quad (47)$$

where dataset D_{t+1} includes the newly evaluated configuration θ^{next} and its smoothed validation loss. This set is then passed to Step 9 to update the surrogate model.

3.9. Update Surrogate

The Bayesian Optimization framework continuously evolves its understanding of the performance surface by integrating new observations into the existing dataset. In the BO-LSTM structure designed for ASD screening classification, each LSTM configuration and its corresponding validation loss provide crucial insights for refining the surrogate model. The most recent configuration-loss pair, obtained from Step 8, is appended to the existing empirical dataset. This augmented set forms the new base for retraining the Gaussian Process surrogate, which enables dynamic recalibration in each iteration of the optimization loop.

$$D_{new} = D_{prev} \cup \{(\theta^{(t+1)}, \bar{L}(\theta^{(t+1)}))\} \quad (48)$$

where D_{new} is the updated training dataset of the surrogate. The new observation includes the configuration $\theta^{(t+1)}$ and its associated smoothed validation loss $\bar{L}$, forming the extended dataset used for posterior reestimation.

The surrogate operates on a kernel matrix derived from the configuration space. This matrix must be re-expanded with each new observation to accommodate the additional configuration. The kernel function measures the similarity between the latest point and all previous entries. This re-expansion ensures that the Gaussian Process continues representing the correlation structure of the complete and current set of LSTM configurations being explored for ASD prediction.

$$K_{t+1}[i,j] = k(\theta^{(i)}, \theta^{(j)}) \quad \forall i,j \in \{1,\dots,t+1\} \quad (49)$$

This expanded kernel matrix K_{t+1} accounts for all configurations from iteration 1 through iteration $t+1$. The function κ computes pairwise covariance using the predefined kernel, typically the Matérn or squared exponential kernel.

Once the kernel matrix is updated, the Gaussian Process must recalculate its predictive mean function to reflect the augmented evidence. The recalibrated posterior mean defines the expected validation loss across the entire configuration space, factoring in historical and newly acquired training results. In BO-LSTM, such recalibration is essential to adaptively recognize emerging regions of interest where configurations yield lower classification loss over the ASD screening dataset.

$$\mu_{t+1}(\theta) = k_{t+1}^{\intercal}(K_{t+1} + \sigma^2 I)^{-1} y_{t+1} \quad (50)$$

The posterior mean $\mu_{t+1}(\theta)$ is now computed using the extended kernel vector K_{t+1}, the updated kernel matrix K_{t+1}, and the new target vector y_{t+1}. This output allows acquisition values in the next iteration to reflect the most recent learning outcome.

The Gaussian Process surrogate also recalculates its uncertainty measure for each configuration. After integrating the latest training observation, the updated posterior variance quantifies the model's confidence in its predictions. This step is instrumental in BO-LSTM for ASD prediction, where minor changes in hyperparameters may produce significant variations in validation loss, and maintaining accurate uncertainty modelling is critical for acquisition design

$$\sigma_{t+1}^2(\theta) = k(\theta,\theta) - k_{t+1}^{\intercal}(K_{t+1} + \sigma^2 I)^{-1} k_{t+1} \quad (51)$$

The posterior variance $\sigma_{t+1}^2(\theta)$ is derived from the covariance between the target configuration and itself, adjusted by its interaction with the updated training set. This value becomes a central component of acquisition function recalculations in Step 10.

After computing the new predictive mean and variance, the scores across all candidate configurations are normalized. This normalization ensures that acquisition values are calculated over a bounded and stable range in the next iteration. In the BO-LSTM pipeline, such normalization eliminates bias introduced by extreme fluctuations in loss values, which are common in small and noisy screening datasets for ASD classification.

$$s_{t+1}(\theta) = \frac{\mu_{t+1}(\theta) - min(\mu_{t+1})}{max(\mu_{t+1}) - min(\mu_{t+1})} \quad (52)$$

This normalized score $s_{t+1}(\theta)$ reflects the relative utility of each configuration, enabling consistent ranking and selection in subsequent acquisition optimization.

With each surrogate update, internal hyperparameters of the kernel function such as length scale and signal variance must be returned to reflect the updated data landscape. This process is performed by re-optimizing the log marginal likelihood, which evaluates how well the Gaussian Process explains the newly extended dataset. This operation strengthens the surrogate's ability to extrapolate meaningfully in unexplored regions of the hyperparameter space.

$$logp(y_{t+1}|\Theta_{t+1}) = -\frac{1}{2}y_{t+1}^{\mathrm{I}}(K_{t+1} + \sigma^2 I)^{-1}y_{t+1} - \frac{1}{2}log\left|(K_{t+1} + \sigma^2 I) - \frac{t+1}{2}log2\pi\right| \quad (53)$$

This marginal log-likelihood represents the model fit to all observed LSTM configurations and their associated loss outcomes, now extended through iteration $t + 1$.

3.10. Select Optimal Config

The Bayesian Optimization cycle continues until a defined stopping condition is satisfied. This condition can be based on a fixed number of iterations, a minimal expected improvement, or stability in loss reductions. For BO-LSTM applied to ASD classification over screening datasets, stability is a preferred criterion since drastic improvements in loss often plateau after multiple evaluations. Once convergence is confirmed, the next phase involves scanning the complete configuration-loss record to isolate the best-performing setup. This setup corresponds to the configuration producing the lowest validation loss across all executed trials.

$$\theta^* = \arg\min_{\theta^{(i)} \in D_T} \bar{L}(\theta^{(i)}) \quad (54)$$

where θ^* represents the final optimal selection. The search is conducted over the whole dataset D_T, which contains all evaluated configurations and their averaged losses. The notation $\bar{L}$ ensures that repeated training trials mitigate stochastic noise.

Each configuration evaluated during the Bayesian Optimization loop can be assigned a ranking based on its final validation loss. This ranking process ensures that all candidates are compared on identical metrics and processed through uniform training-validation splits. Empirical ranking reinforces fairness and interpretability in the configuration selection step for ASD classification, especially in cases where multiple configurations yield marginally different losses. This sorting also helps visualize performance patterns across the hyperparameter space.

$$R\left(\theta^{(i)}\right) = Rank\left(\left\{\bar{L}\left(\theta^{(1)}\right), \dots, \bar{L}\left(\theta^{(T)}\right)\right\}, \theta^{(i)}\right) \quad (55)$$

The function $R\left(\theta^{(i)}\right)$ assigns a rank to each configuration $\theta^{(i)}$ based on its corresponding loss $\bar{L}$. Lower ranks indicate superior performance. This information can be visualized or stored for audit and model reproducibility tracking.

After determining the best configuration through empirical minimization and rank validation, this vector is fixed for final model deployment. This finalized LSTM hyperparameter vector contains specific values for learning rate, dropout ratio, hidden units, and batch size that demonstrate superior ability to classify ASD cases from non-imaging screening inputs. The configuration is extracted from the history and serves as the blueprint for training the definitive version of the BO-LSTM model.

$$\theta^* = [\eta^*, \delta^*, u^*, b^*] \quad (56)$$

where vector θ^* includes the optimal learning rate η^*, dropout δ^*, unit count u^*, and batch size b^*. These values are not theoretical estimates but empirically validated selections from probabilistic modelling and surrogate-guided evaluations in earlier steps.

To enable transparency and future benchmarking, the selected configuration's

validation performance is logged alongside other key metrics such as convergence time, total configurations evaluated, and the range of observed losses. This structured logging allows future evaluations and comparisons with optimization techniques or base models. In the context of ASD screening analysis, this record provides an auditable trace of how the optimal configuration was derived and substantiates the final model's validity.

$$M_{final} = (\theta^*, \bar{L}(\theta^*), T, D_T) \quad (57)$$

where tuple M_{final} contains the final configuration, its observed average loss, the total number of iterations T, and the complete dataset of all configurations evaluated. This package encapsulates the outcome of the Bayesian Optimization process.

3.11. Overall Procedure for BO-LSTM Framework

The BO-LSTM framework's complete process combines all the steps into one organized flow for predicting ASD from screening data. It starts by defining a specific range for essential parameters like learning rate, dropout, LSTM units, and batch size. A Gaussian Process estimates which settings might give better results based on a few tested combinations. These early results help guide the next set of parameters to try, using a method that balances learning from past outcomes and exploring new options. The LSTM model is trained for each chosen configuration using the sequence of responses from screening questionnaires. After each training round, results are returned to the system to improve its choices in the next round. This cycle continues until no further improvements are found or the set number of attempts is reached. The best configuration during these trials is then selected to build the final model. This procedure ensures the LSTM is appropriately tuned, handles real-world behavioral data effectively, and supports early detection of ASD with high accuracy and minimal manual adjustment.

Algorithm: BO-LSTM

Input:

- ASD screening dataset $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
- Defined search space H for LSTM hyperparameters
- Maximum iteration count T or convergence threshold ϵ

Output:

- Optimal LSTM hyperparameter configuration θ^*
- Trained LSTM model M_{θ^*} for ASD prediction

Procedure:

1. **Initialize Surrogate:** Construct a Gaussian Process surrogate model with a chosen kernel to approximate validation loss over the hyperparameter space.
2. **Define Search Space:** Specify the domain H for hyperparameters, including learning rate, dropout, hidden units, and batch size.
3. **Sample Initial Points:** Generate k initial configurations using Latin Hypercube Sampling and evaluate each on the LSTM model to obtain validation losses.
4. **Train LSTM Samples:** Train LSTM models for all initial configurations and record smoothed validation losses by averaging over multiple runs.
5. **Fit a Gaussian Process:** Fit the surrogate model using observed configuration-loss pairs and optimize its internal parameters via marginal likelihood.
6. **Compute Acquisition:** Calculate acquisition function (e.g., Expected Improvement) using the surrogate's posterior mean and variance.
7. **Select Next Point:** Identify the configuration that maximizes the acquisition function and satisfies all domain constraints and confidence thresholds.
8. **Train with Suggestion:** Train an LSTM using the selected configuration and evaluate its validation loss across multiple trials.
9. **Update Surrogate:** Update the surrogate dataset with the new configuration-loss pair and recompute the surrogate model.
10. **Select Optimal Config:** Once the convergence or iteration limit is reached, select the configuration with the lowest observed loss from the complete evaluation history.

3.11.1. Advantages of BO-LSTM

The BO-LSTM framework offers a structured and intelligent approach to optimizing deep learning models for ASD prediction using screening data. By combining sequential modeling with automated Bayesian tuning, the framework

reduces the reliance on manual experimentation and improves generalization in behaviorally diverse scenarios. Its design supports scalable deployment and efficient training, making it suitable for real-world diagnostic applications across varied populations. The significant advantages of BO-LSTM are:

- **Automated Hyperparameter Tuning:** Reduces manual effort by using Bayesian Optimization to find optimal LSTM settings without exhaustive search.
- **Adaptation to Sequential Screening Data:** Effectively captures temporal patterns in structured questionnaire responses, improving prediction quality.
- **Improved Generalization Across Age Groups:** Learns from diverse behavioural traits, enhancing consistency across toddlers, children, adolescents, and adults.
- **Reduced Overfitting in Low-Dimensional Data:** Probabilistic selection and averaging stabilize learning in datasets with limited features and class imbalance.
- **Resource-Efficient Training Process:** Minimizes the number of training runs by focusing only on the most promising configurations, saving time and computation.

3.11.2. Difference between LSTM and BO-LSTM

While Long Short-Term Memory (LSTM) networks are well-suited for modeling sequential data, their effectiveness heavily depends on carefully selecting hyperparameters such as learning rate, dropout, and hidden units. Manually tuning these settings can be time-consuming and may not consistently yield optimal results, especially when applied to behaviorally diverse datasets like ASD screening records. To address these limitations, the BO-LSTM framework extends the standard LSTM by integrating Bayesian Optimization, enabling automatic and efficient exploration of hyperparameter configurations. The following table outlines the key differences between the conventional LSTM model and the proposed BO-LSTM approach.

Table 1. Difference Between LSTM And BO-LSTM

Aspect	***LSTM***	***BO-LSTM***
Hyperparameter Tuning	Manual or heuristic-based	Automated using Bayesian Optimization
Optimization Strategy	No built-in optimization mechanism	Uses surrogate modeling and acquisition-driven parameter selection
Training Efficiency	It may involve redundant training trials	Focuses on high-potential configurations, reducing training cost
Model Adaptability	Sensitive to initial settings	Adapts dynamically to data and parameter landscape
Generalization	Prone to overfitting on small or imbalanced data	Maintains robust performance across diverse input patterns

4. DATASET

The Autism Screening Dataset contains 6075 records and 20 structured attributes. The dataset was developed by Dr. Fadi Fayez using the ASD Tests mobile application (ASDtests.com) for early screening of autism traits and is publicly available on Kaggle (https://www.kaggle.com/datasets/fabdelja/asd-screening-data-toddler-child-adoles-adult).It combines behavioural screening responses from four distinct age groups: Toddler, Child, adolescent, and Adult. Q-CHAT-10 was used for the Toddler group, and AQ-10 short-form questionnaires were applied to the remaining categories. Each entry represents a completed, structured screening session the individual or observer reports. The dataset is fully anonymized and ethically shared, making it safe for academic and research use. Its consistent structure supports population-wide behavioural studies focused on non-clinical trait analysis. Including multi-age data allows for comparative trend exploration and trait expression across developmental stages. This dataset is a robust, non-invasive resource for autism-related behavioural screening research focusing on early identification.

Table 2: Feature Description

Feature Name	*Description*	*Data Type / Format*
ID	Record number uniquely assigned to each entry	Categorical (String)
A1_Score	Response to item 1 of the assessment	Binary (Yes/No)
A2_Score	Feedback on the second item from the behavioural checklist	Binary (Yes/No)
A3_Score	Recorded answer for item 3	Binary (Yes/No)
A4_Score	Response reflecting the social observation	Binary (Yes/No)
A5_Score	Score related to attention and interest	Binary (Yes/No)
A6_Score	Input indicating behaviour under peer influence	Binary (Yes/No)
A7_Score	Score reflecting communication irregularities	Binary (Yes/No)
A8_Score	The entry focused on adaptability or rigidity	Binary (Yes/No)
A9_Score	Reaction to structured versus unstructured environments	Binary (Yes/No)
A10_Score	The final item in the behavioural checklist	Binary (Yes/No)
age	Participant's declared age	Numeric (Years/Float)
gender	Categorical entry for sex	Categorical (Male/Female)
ethnicity	Ethnic or cultural identity reported	Categorical (Free-text)
jaundice	Neonatal health condition status	Binary (Yes/No)
family_mem_with_ASD	Presence of ASD diagnosis in the family line	Binary (Yes/No)
who_completed_the_test	Test responder's role or identity	Categorical (Free-text)
country_of_res	The nation of residence mentioned	Categorical (Free-text)
used_app_before	Declares app usage history	Binary (Yes/No)
result	ASD suspicion flag from the screening	Binary (Yes/No)
age_desc	Participant's age grouping	Categorical (Defined Set)
relation	Nature of association between subject and responder	Categorical (Free-text)

5. RESULTS AND DISCUSSIONS

The evaluation focuses on the predictive performance of the BO-LSTM model compared to baseline classifiers across multiple diagnostic metrics. Sensitivity, specificity, accuracy, Matthews Correlation Coefficient, threat score, and Fowlkes–Mallows Index are used to measure how well each model identifies autism-related traits from screening data. Results are analyzed with attention to model robustness, consistency across age groups, and handling of behavioural variability. Using Bayesian Optimization within the BO-LSTM architecture contributes to stable generalization by reducing overfitting and improving configuration efficiency. Comparative findings reveal the strength of BO-LSTM in learning from structured screening responses, particularly in conditions where trait expression is subtle or overlapping. The analysis also considers the balance between true positive recognition and false positive control, which is essential in real-world ASD screening applications.

5.1. Sensitivity Analysis of BO-LSTM

Figure 1 compares the sensitivity of BO-LSTM with SVM and BDML-MDCASD classifiers, drawing upon the values outlined in Table 2. Sensitivity, representing the true positive rate, is critical in ASD screening where missed cases carry significant clinical impact. SVM demonstrates low sensitivity (61.490%) due to its limited capacity to model contextual dependencies across behavioural traits. BDML-MDCASD shows improvement but is

Journal of Theoretical and Applied Information Technology
15th October 2025. Vol.103. No.19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

affected by latent representation distortion and lack of temporal structuring. BO-LSTM, reaching 76.858%, benefits from its gated memory design, which preserves sequential patterns indicative of ASD. Bayesian optimization further contributes by tuning model parameters specific to the screening context. These structural enhancements enable more reliable identification of true ASD-positive cases within age-diverse populations.

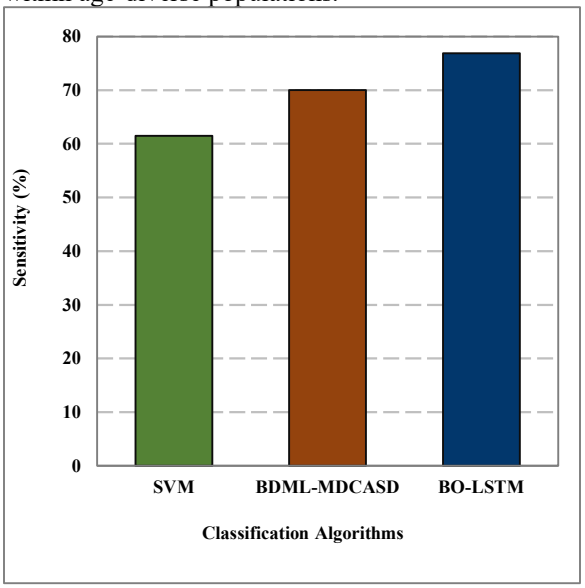

Figure 1. Sensitivity of BO-LSTM against State-of-the-Art Classification Algorithms

Table 2. Sensitivity Result of BO-LSTM State-of-the-Art Classification Algorithms

Classification Algorithms	Specificity (%)
SVM	51.934
BDML-MDCASD	68.138
BO-LSTM	71.800

5.2. Specificity Analysis of BO-LSTM

Figure 2 presents a comparative specificity evaluation across BO-LSTM, SVM, and BDML-MDCASD, with detailed metrics reported in Table 3. Specificity, measuring true negative identification, is vital in reducing false positives in ASD screening tasks. The SVM classifier, scoring 51.934%, fails to mitigate boundary misalignment in feature space due to its hard-margin structure and absence of noise-adaptive filters. BDML-MDCASD achieves 68.138%, though the interpretive loss in AE compression and BOA's context-agnostic tuning dilutes its precision. BO-LSTM registers 71.800% specificity by aligning forget gates with non-ASD behavioural regularities and modulating noise-sensitive hyperparameters through Bayesian posterior updates. This integration refines model discrimination against non-ASD instances with higher fidelity.

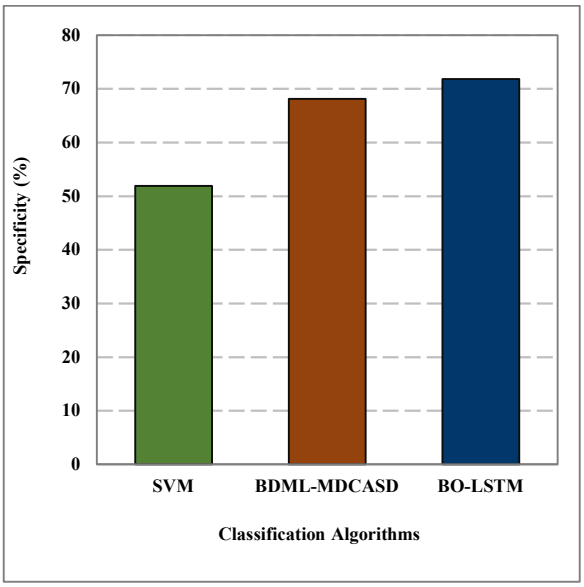

Figure 2. Specificity of BO-LSTM against State-of-the-Art Classification Algorithms

Table 3. Specificity Result of BO-LSTM vs State-of-the-Art Models

Classification Algorithms	Sensitivity (%)
SVM	61.490
BDML-MDCASD	70.023
BO-LSTM	76.858

5.3. Classification Accuracy Analysis of BO-LSTM

Figure 3 illustrates the classification accuracy of BO-LSTM against state-of-the-art models, with tabulated results in Table 4. Classification accuracy assesses global prediction performance, aggregating true positives and true negatives. The SVM model, constrained by rigid kernel mapping and static feature interpretation, yields 56.365%. BDML-MDCASD, though enhanced via heuristic feature selection and representation compression, attains 69.089% but suffers from interpretability gaps and parameter tuning instability. BO-LSTM, achieving 74.375%, leverages the synergy between temporal memory

flow and Bayesian-informed structural calibration. Its recurrent dynamics facilitate retention of questionnaire order, while probabilistic search optimizes model depth and connectivity. These layered advancements enable a holistic capture of ASD and non-ASD profiles, elevating classification integrity across the screening spectrum.

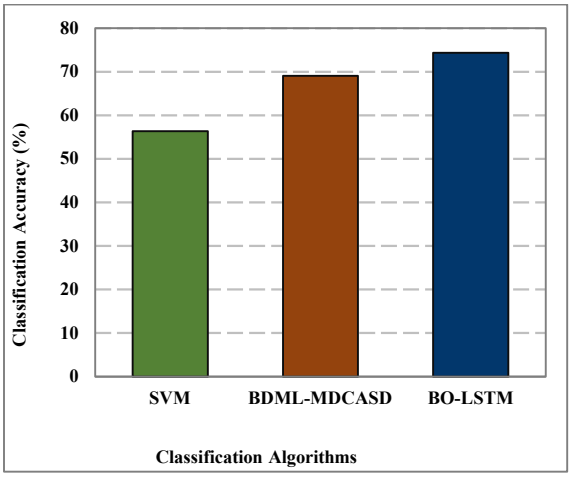

Figure 3. Classification Accuracy Of BO-LSTM Against State-Of-The-Art Classifiers

Table 4. Accuracy Result Of BO-LSTM And Competing Models

Classification Algorithms	Classification Accuracy (%)
SVM	56.365
BDML-MDCASD	69.089
BO-LSTM	74.375

5.4. Matthews Correlation Coefficient Analysis of BO-LSTM

Figure 4 visualizes the Matthews Correlation Coefficient (MCC) of BO-LSTM compared to SVM and BDML-MDCASD, with corresponding numeric values in Table 5. MCC evaluates the quality of binary classifications by incorporating all confusion matrix components, providing a balanced view even under class imbalance—a known characteristic of ASD datasets.

Table 5. Matthews Correlation Coefficient Result of BO-LSTM Compared to Other Methods

Classification Algorithms	Matthews Correlation Coefficient (%)
SVM	13.438
BDML-MDCASD	38.169
BO-LSTM	48.736

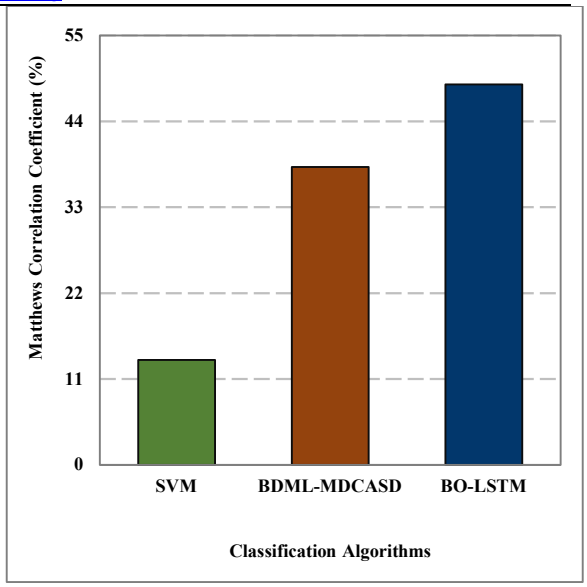

Figure 4. Matthews Correlation Coefficient of BO-LSTM against State-of-the-Art Classification Algorithms

SVM, with an MCC of 13.438%, performs poorly due to its sensitivity to skewed class distribution and inability to integrate feature interdependencies. BDML-MDCASD exhibits moderate improvement (38.169%) but inherits limitations from its disjoint optimization structure; the BOA layer fails to adapt to class overlap fully, and the Autoencoder reduces behavioural interpretability. BO-LSTM, at 48.736%, shows a stronger correlation between predicted and actual classifications. This is attributed to its memory retention of sequential traits, probabilistic tuning of internal states, and capacity to modulate relevance across mixed-type screening features—resulting in a structurally balanced decision function resilient to data asymmetry.

5.5. Threat Score Analysis of BO-LSTM

Figure 5 showcases the threat score performance of BO-LSTM compared to SVM and BDML-MDCASD, with numerical details provided in Table 6. Threat score, also known as critical success index, quantifies the proportion of correctly predicted positive instances relative to all predicted and actual positives—making it particularly suited for ASD screening tasks where positive case detection is a priority.

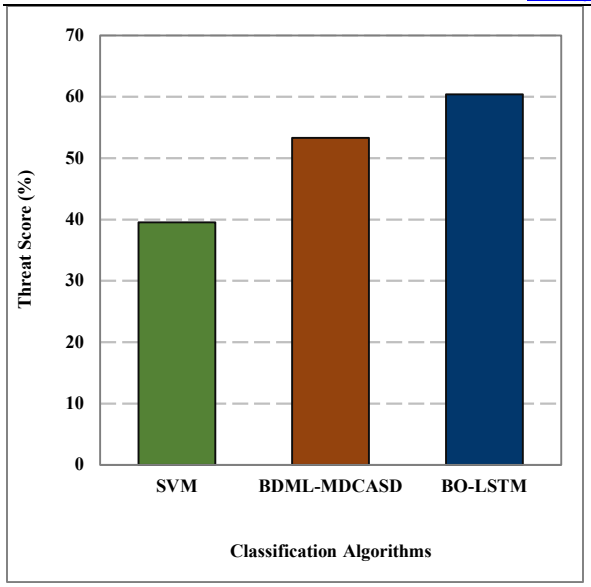

Figure 5. Threat Score Of BO-LSTM Against State-Of-The-Art Classifiers

SVM records a lower threat score (39.518%) primarily because it lacks adaptive mechanisms to resolve borderline behavioural traits, often misclassifying subtle ASD patterns. BDML-MDCASD fares better (53.333%) yet remains affected by its inability to contextualize trait relevance within its feature-to-decision pathway, as its layered optimizers operate in isolation. BO-LSTM, achieving the highest threat score (60.426%), demonstrates superior alignment between predicted and true ASD-positive profiles. This is driven by its gated architecture that retains progression cues and Bayesian hyperparameter sampling that fine-tunes the decision boundary following screening-specific uncertainty.

Table 6. Threat Score Result Of BO-LSTM Against State-Of-The-Art Classifiers

Classification Algorithms	Threat Score (%)
SVM	39.518
BDML-MDCASD	53.333
BO-LSTM	60.426

5.6. Fowlkes–Mallows Index Analysis of BO-LSTM

Figure 6 illustrates the Fowlkes–Mallows Index (FMI) performance of BO-LSTM compared with SVM and BDML-MDCASD, with supporting numerical values in Table 7. FMI evaluates the geometric mean between precision and recall, making it a reliable indicator of balance in-class assignment—critical in ASD classification where both false positives and false negatives must be minimized.

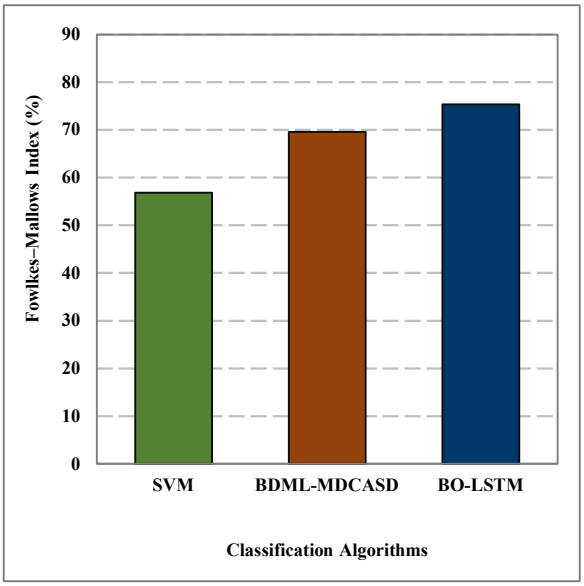

Figure 6. FMI Of BO-LSTM Against State-Of-The-Art Classification Algorithms

SVM's FMI of 56.826% is directly impacted by its inability to accommodate semantically overlapping behavioural features, as it applies uniform separation rules to symptomatically nuanced data. BDML-MDCASD, reaching 69.566%, is structurally stronger, yet the absence of feature-level interpretability and lack of joint optimization between AE and BOA layers restricts its decision coherence. BO-LSTM, registering 75.347%, achieves superior balance by encoding temporal cues through memory units and refining prediction boundaries via Bayesian parameter sampling. These mechanisms work in tandem to stabilize classification behaviour across high-variability screening records.

Table *7. Fowlkes–Mallows Index Result Against State-Of-The-Art Classification Algorithms*

Classification Algorithms	Fowlkes–Mallows Index (%)
SVM	56.826
BDML-MDCASD	69.566
BO-LSTM	75.347

5.7. Interpretation of Findings and Practical Implications

The comparative analysis highlights that the proposed model consistently surpasses baseline

classifiers across all evaluated metrics, indicating its robustness in handling behavioural diversity and noisy screening inputs. This improvement suggests that constraint-aware reinforcement structures, when adapted for non-imaging ASD datasets, can mitigate instability and bias commonly observed in existing methods. The performance gains are not only statistically significant but also practically relevant, as they demonstrate the model's ability to maintain predictive stability across varied demographic profiles and diagnostic stages. By sustaining high accuracy alongside balanced sensitivity and specificity, the approach addresses the long-standing challenge of overfitting to narrow population segments. These findings imply that reinforcement learning models, guided by structured constraints, can form the foundation for deployable ASD screening systems in community and clinical settings.

5.8. Potential Real-World Implementation

The proposed methodology holds promise for integration into digital health tools designed for early ASD screening. In a practical setting, the model could be embedded into mobile or web-based platforms to analyse responses from structured behavioural questionnaires and produce preliminary risk scores. Such systems may assist teachers, caregivers, and primary care providers in identifying children who require specialist evaluation. Its tolerance for incomplete and imbalanced data suggests potential use in community health programs, where screening conditions are often variable. With further validation on larger and more diverse datasets, the approach could be adapted for integration with existing electronic health record systems, supporting longitudinal tracking of developmental profiles. These envisioned applications outline how the research could transition from an experimental framework to a practical decision-support tool in ASD identification.

6. CONCLUSION

The proposed framework demonstrates that embedding constraint-driven optimization into deep reinforcement learning enables robust ASD prediction from non-imaging screening datasets. By addressing instability, class imbalance, and uncertainty, the model achieves consistent performance across varied diagnostic stages and demographic profiles. These findings indicate that reinforcement learning models with structured constraints can form a reliable basis for deployable ASD screening systems, particularly in settings where costly neuroimaging or specialist evaluations are not feasible. In practical terms, the methodology could be integrated into mobile or web-based screening tools to process questionnaire data and deliver preliminary risk scores, guiding timely referrals for specialist evaluation. Its resilience to incomplete and imbalanced inputs suggests potential applicability in community health programs and low-resource environments, once validated on broader datasets. Future research should focus on addressing the current limitation of dataset diversity by evaluating the model across larger, multi-regional, and multi-lingual screening datasets to ensure cultural and linguistic generalisation. Integration with multi-modal data sources—such as speech patterns, eye-tracking, and caregiver interviews—could enrich feature space and improve early detection sensitivity. Moreover, adapting the framework for active learning would allow the system to refine predictions continuously as new labelled data become available. Exploring explainable AI modules within the architecture could further enhance interpretability, making the model more transparent and trusted by clinicians. These directions, guided by gaps in the literature and limitations in the present work, pave the way for transitioning the approach from a research prototype to a widely adopted screening solution.

REFERENCES

[1] M. Pavethra and M. and Uma Devi, "A cross layer graphical neural network based convolutional neural network framework for image dehazing," *Automatika*, vol. 65, no. 3, pp. 1139–1153, Jul. 2024, doi: 10.1080/00051144.2024.2346964.

[2] H.-D. Li, C. Deng, X.-Q. Zhang, and C.-X. Lin, "A Gene Set-Integrated Approach for Predicting Disease-Associated Genes," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 20, no. 6, pp. 3440–3450, 2023, doi: 10.1109/TCBB.2022.3214517.

[3] D. P. Kavadi, V. R. R. Chirra, P. R. Kumar, S. B. Veesam, S. Yeruva, and L. K. Pappala, "A Hybrid Machine Learning Model for Accurate Autism Diagnosis," *IEEE Access*, vol. 12, pp. 194911–194921, 2024, doi: 10.1109/ACCESS.2024.3520009.

[4] 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.

[5] B. Farrow, J. Soo-Yeon, and S. and Jayarathna,

"A microservices architecture for processing large electroencephalogram studies," *Int. J. Comput. Appl.*, vol. 47, no. 3, pp. 329–338, Mar. 2025, doi: 10.1080/1206212X.2025.2450247.

[6] Y. Ma, W. Cui, J. Liu, Y. Guo, H. Chen, and Y. Li, "A Multi-Graph Cross-Attention-Based Region-Aware Feature Fusion Network Using Multi-Template for Brain Disorder Diagnosis," *IEEE Trans. Med. Imaging*, vol. 43, no. 3, pp. 1045–1059, 2024, doi: 10.1109/TMI.2023.3327283.

[7] Q. Dong, H. Cai, Z. Li, J. Liu, and B. Hu, "A Multiview Brain Network Transformer Fusing Individualized Information for Autism Spectrum Disorder Diagnosis," *IEEE J. Biomed. Heal. Informatics*, vol. 28, no. 8, pp. 4854–4865, 2024, doi: 10.1109/JBHI.2024.3396457.

[8] G. Matthews, C. Ryon, D. L. S. Erika P., F. Irene Y., and S. A. and Mouloua, "A new era for stress research: supporting user performance and experience in the digital age," *Ergonomics*, vol. 68, no. 6, pp. 913–946, Jun. 2025, doi: 10.1080/00140139.2024.2425953.

[9] W. Huang *et al.*, "A novel brain inception neural network model using EEG graphic structure for emotion recognition," *Brain-Apparatus Commun. A J. Bacomics*, vol. 2, no. 1, p. 2222159, Dec. 2023, doi: 10.1080/27706710.2023.2222159.

[10] K. Harini and S. and Uma Maheswari, "A novel static and dynamic hand gesture recognition using self organizing map with deep convolutional neural network," *Automatika*, vol. 64, no. 4, pp. 1128–1140, Oct. 2023, doi: 10.1080/00051144.2023.2251229.

[11] 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. 15739–15750, 2025, doi: 10.1109/ACCESS.2025.3531979.

[12] Z. Zeng *et al.*, "A Robust Gaze Estimation Approach via Exploring Relevant Electrooculogram Features and Optimal Electrodes Placements," *IEEE J. Transl. Eng. Heal. Med.*, vol. 12, pp. 56–65, 2024, doi: 10.1109/JTEHM.2023.3320713.

[13] Q. Xia, C. Thomas K. F., and X. and Li, "A scoping review of BCIs for learning regulation in mainstream educational contexts," *Behav. Inf. Technol.*, vol. 43, no. 10, pp. 2096–2117, Jul. 2024, doi: 10.1080/0144929X.2023.2241559.

[14] M. Bondioli, C. Stefano, K. Alexander, and S. and Pelagatti, "A survey on technological tools and systems for diagnosis and therapy of autism spectrum disorder," *Human–Computer Interact.*, vol. 39, no. 3–4, pp. 145–173, Jul. 2024, doi: 10.1080/07370024.2023.2242345.

[15] H. Zhu, J. Wang, Y.-P. Zhao, M. Lu, and J. Shi, "Contrastive Multi-View Composite Graph Convolutional Networks Based on Contribution Learning for Autism Spectrum Disorder Classification," *IEEE Trans. Biomed. Eng.*, vol. 70, no. 6, pp. 1943–1954, 2023, doi: 10.1109/TBME.2022.3232104.

[16] W. Zhou, M. Yang, J. Tang, J. Wang, and B. Hu, "Gaze Patterns in Children With Autism Spectrum Disorder to Emotional Faces: Scanpath and Similarity," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 32, pp. 865–874, 2024, doi: 10.1109/TNSRE.2024.3361935.

[17] J. Zhao *et al.*, "Electrophysiological Abnormalities Associated With Sustained Attention in Children With Attention Deficit Hyperactivity Disorder and Autism Spectrum Disorder," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 33, pp. 1785–1795, 2025, doi: 10.1109/TNSRE.2025.3564608.

[18] D. Jayaraj, J. Ramkumar, M. Lingaraj, and B. Sureshkumar, "AFSORP: Adaptive Fish Swarm Optimization-Based Routing Protocol for Mobility Enabled Wireless Sensor Network," *Int. J. Comput. Networks Appl.*, vol. 10, no. 1, pp. 119–129, 2023, doi: 10.22247/ijcna/2023/218516.

[19] J. Ramkumar, R. Karthikeyan, and V. Valarmathi, "Alpine Swift Routing Protocol (ASRP) for Strategic Adaptive Connectivity Enhancement and Boosted Quality of Service in Drone Ad Hoc Network (DANET)," *Int. J. Comput. Networks Appl.*, vol. 11, no. 5, pp. 726–748, 2024, doi: 10.22247/ijcna/2024/45.

[20] S. P. Priyadharshini, F. Nirmala Irudayam, and J. Ramkumar, "An Unique Overture of Plithogenic Cubic Overset, Underset and Offset," in *Studies in Fuzziness and Soft Computing*, vol. 435, 2025, pp. 139–156. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-105001675443&doi=10.1007%2F978-3-031-78505-4_7&partnerID=40&md5=e9def8c6a233de4fbf8f1549ad72027f

[21] R. Jaganathan, S. Mehta, and R. Krishan, *Bio-Inspired intelligence for smart decision-making*, vol. i. 2024. doi: 10.4018/9798369352762.

[22] J. Ramkumar, R. Vadivel, and B. Narasimhan, "Constrained Cuckoo Search Optimization Based Protocol for Routing in Cloud Network," *Int. J. Comput. Networks Appl.*, vol. 8, no. 6, pp.

795–803, 2021, doi: 10.22247/ijcna/2021/210727.

[23] J. Ramkumar and R. Vadivel, "CSIP—cuckoo search inspired protocol for routing in cognitive radio ad hoc networks," in *Advances in Intelligent Systems and Computing*, Springer Verlag, 2017, pp. 145–153. doi: 10.1007/978-981-10-3874-7_14.

[24] N. K. Ojha, A. Pandita, and J. Ramkumar, "Cyber security challenges and dark side of AI: Review and current status," in *Demystifying the Dark Side of AI in Business*, IGI Global, 2024, pp. 117–137. doi: 10.4018/979-8-3693-0724-3.ch007.

[25] J. Ramkumar, C. Kumuthini, B. Narasimhan, and S. Boopalan, "Energy Consumption Minimization in Cognitive Radio Mobile Ad-Hoc Networks using Enriched Ad-hoc On-demand Distance Vector Protocol," *2022 Int. Conf. Adv. Comput. Technol. Appl. ICACTA 2022*, pp. 1–6, Mar. 2022, doi: 10.1109/ICACTA54488.2022.9752899.

[26] S. P. Geetha, N. M. S. Sundari, J. Ramkumar, and R. Karthikeyan, "Energy Efficient Routing In Quantum Flying Ad Hoc Network (Q-FANET) Using Mamdani Fuzzy Inference Enhanced Dijkstra's Algorithm (MFI-EDA)," *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 9, pp. 3708–3724, 2024, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85197297302&partnerID=40&md5=72d51668bee6239f09a59d2694df67d6

[27] M. P. Swapna, J. Ramkumar, and R. Karthikeyan, "Energy-Aware Reliable Routing with Blockchain Security for Heterogeneous Wireless Sensor Networks," in *Lecture Notes in Networks and Systems*, V. Goar, M. Kuri, R. Kumar, and T. Senjyu, Eds., Springer Science and Business Media Deutschland GmbH, 2025, pp. 713–723. doi: 10.1007/978-981-97-6106-7_43.

[28] B. Suchitra, R. Karthikeyan, J. Ramkumar, and V. Valarmathi, "Enhancing Recurrent Neural Network Performance For Latent Autoimmune Diabetes Detection (LADA) Using Exocoetidae Optimization," *J. Theor. Appl. Inf. Technol.*, vol. 103, no. 5, pp. 1645–1667, 2025, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-105000948603&partnerID=40&md5=66c8f111b153fed68b3d0ea9c88c411e

[29] B. Suchitra, J. Ramkumar, and R. Karthikeyan, "Frog Leap Inspired Optimization-Based Extreme Learning Machine for Accurate Classification of Latent Autoimmune Diabetes in Adults (LADA)," *J. Theor. Appl. Inf. Technol.*, vol. 103, no. 2, pp. 472–494, 2025, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85217140979&partnerID=40&md5=9540433c16d5ff0f6c2de4b8c43a4812

[30] J. Ramkumar and R. Vadivel, "Improved frog leap inspired protocol (IFLIP) – for routing in cognitive radio ad hoc networks (CRAHN)," *World J. Eng.*, vol. 15, no. 2, pp. 306–311, 2018, doi: 10.1108/WJE-08-2017-0260.

[31] J. Ramkumar and R. Vadivel, "Improved Wolf prey inspired protocol for routing in cognitive radio Ad Hoc networks," *Int. J. Comput. Networks Appl.*, vol. 7, no. 5, pp. 126–136, 2020, doi: 10.22247/ijcna/2020/202977.

[32] R. Jaganathan, S. Mehta, and R. Krishan, *Intelligent Decision Making Through Bio-Inspired Optimization*. IGI Global, 2024. doi: 10.4018/979-8-3693-2073-0.

[33] R. Jaganathan and R. Vadivel, "Intelligent Fish Swarm Inspired Protocol (IFSIP) for Dynamic Ideal Routing in Cognitive Radio Ad-Hoc Networks," *Int. J. Comput. Digit. Syst.*, vol. 10, no. 1, pp. 1063–1074, 2021, doi: 10.12785/ijcds/100196.

[34] J. Ramkumar, S. S. Dinakaran, M. Lingaraj, S. Boopalan, and B. Narasimhan, "IoT-Based Kalman Filtering and Particle Swarm Optimization for Detecting Skin Lesion," in *Lecture Notes in Electrical Engineering*, K. Murari, S. Kamalasadan, and N. P. Padhy, Eds., Springer Science and Business Media Deutschland GmbH, 2023, pp. 17–27. doi: 10.1007/978-981-19-8353-5_2.

[35] J. Ramkumar, B. Varun, V. Valarmathi, D. R. Medhunhashini, and R. Karthikeyan, "Jaguar-Based Routing Protocol (JRP) For Improved Reliability and Reduced Packet Loss in Drone Ad-Hoc Networks (DANET)," *J. Theor. Appl. Inf. Technol.*, vol. 103, no. 2, pp. 696–713, 2025, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85217213044&partnerID=40&md5=e38a375e46cf43c95d6702a3585a7073

[36] J. Ramkumar and D. Ravindran, "Machine learning and robotics in urban traffic flow optimization with graph neural networks and reinforcement learning," in *Machine Learning and Robotics in Urban Planning and Management*, 2025, pp. 83–104. [Online].

Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-105000106746&doi=10.4018%2F979-8-3693-9410-6.ch005&partnerID=40&md5=f647b3741afce4400893c2913f2bbf55

[37] S. P. Priyadharshini and J. Ramkumar, "Mappings Of Plithogenic Cubic Sets," *Neutrosophic Sets Syst.*, vol. 79, pp. 669–685, 2025, doi: 10.5281/zenodo.14607210.

[38] A. Senthilkumar, J. Ramkumar, M. Lingaraj, D. Jayaraj, and B. Sureshkumar, "Minimizing Energy Consumption in Vehicular Sensor Networks Using Relentless Particle Swarm Optimization Routing," *Int. J. Comput. Networks Appl.*, vol. 10, no. 2, pp. 217–230, 2023, doi: 10.22247/ijcna/2023/220737.

[39] V. Valarmathi and J. Ramkumar, "Modernizing Wildfire Management Through Deep Learning and IoT in Fire Ecology," in *Machine Learning and Internet of Things in Fire Ecology*, 2024, pp. 203–229. doi: 10.4018/979-8-3693-7565-5.ch0010.

[40] J. Ramkumar and R. Vadivel, "Multi-Adaptive Routing Protocol for Internet of Things based Ad-hoc Networks," *Wirel. Pers. Commun.*, vol. 120, no. 2, pp. 887–909, Apr. 2021, doi: 10.1007/s11277-021-08495-z.

[41] M. P. Swapna and J. Ramkumar, "Multiple Memory Image Instances Stratagem to Detect Fileless Malware," in *Communications in Computer and Information Science*, S. Rajagopal, K. Popat, D. Meva, and S. Bajeja, Eds., Cham: Springer Nature Switzerland, 2024, pp. 131–140. doi: 10.1007/978-3-031-59100-6_11.

[42] J. Ramkumar, A. Senthilkumar, M. Lingaraj, R. Karthikeyan, and L. Santhi, "Optimal Approach for Minimizing Delays in IoT-Based Quantum Wireless Sensor Networks Using NM-Leach Routing Protocol," *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 3, pp. 1099–1111, 2024, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85185481011&partnerID=40&md5=bf0ff974ceabc0ad58e589b28797c684

[43] J. Ramkumar, R. Karthikeyan, and M. Lingaraj, "Optimizing IoT-Based Quantum Wireless Sensor Networks Using NM-TEEN Fusion of Energy Efficiency and Systematic Governance," in *Lecture Notes in Electrical Engineering*, V. Shrivastava, J. C. Bansal, and B. K. Panigrahi, Eds., Springer Science and Business Media Deutschland GmbH, 2025, pp. 141–153. doi: 10.1007/978-981-97-6710-6_12.

[44] J. Ramkumar, V. Valarmathi, and R. Karthikeyan, "Optimizing Quality of Service and Energy Efficiency in Hazardous Drone Ad-Hoc Networks (DANET) Using Kingfisher Routing Protocol (KRP)," *Int. J. Eng. Trends Technol.*, vol. 73, no. 1, pp. 410–430, 2025, doi: 10.14445/22315381/IJETT-V73I1P135.

[45] R. Jaganathan, K. Rajendran, and P. S. Ponnukumar, "Peregrine Falcon Optimization Routing Protocol (PFORP) for Achieving Ultra-Low Latency and Boosted Efficiency in 6G Drone Ad-Hoc Networks (DANET)," *Int. J. Comput. Digit. Syst.*, vol. 17, no. 1, pp. 1–18, 2025, doi: 10.12785/ijcds/1571111848.

[46] L. Mani, S. Arumugam, and R. Jaganathan, "Performance Enhancement of Wireless Sensor Network Using Feisty Particle Swarm Optimization Protocol," *ACM Int. Conf. Proceeding Ser.*, pp. 1–5, Dec. 2022, doi: 10.1145/3590837.3590907.

[47] R. Jaganathan and V. Ramasamy, "Performance modeling of bio-inspired routing protocols in Cognitive Radio Ad Hoc Network to reduce end-to-end delay," *Int. J. Intell. Eng. Syst.*, vol. 12, no. 1, pp. 221–231, 2019, doi: 10.22266/IJIES2019.0228.22.

[48] R. Jaganathan, S. Mehta, and R. Krishan, "Preface," *Bio-Inspired Intell. Smart Decis.*, pp. xix–xx, 2024, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85195725049&partnerID=40&md5=7a2aa7adc005662eebc12ef82e3bd19f

[49] R. Jaganathan, S. Mehta, and R. Krishan, "Preface," *Intell. Decis. Mak. Through Bio-Inspired Optim.*, pp. xiii–xvi, 2024, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85192858710&partnerID=40&md5=f8f1079e8772bd424d2cdd979e5f2710

[50] R. Vadivel and J. Ramkumar, "QoS-enabled improved cuckoo search-inspired protocol (ICSIP) for IoT-based healthcare applications," in *Incorporating the Internet of Things in Healthcare Applications and Wearable Devices*, IGI Global, 2019, pp. 109–121. doi: 10.4018/978-1-7998-1090-2.ch006.

[51] M. Lingaraj, T. N. Sugumar, C. S. Felix, and J. Ramkumar, "Query aware routing protocol for mobility enabled wireless sensor network," *Int. J. Comput. Networks Appl.*, vol. 8, no. 3, pp. 258–267, 2021, doi: 10.22247/ijcna/2021/209192.

[52] J. Ramkumar, K. S. Jeen Marseline, and D. R. Medhunhashini, "Relentless Firefly Optimization-Based Routing Protocol (RFORP) for Securing Fintech Data in IoT-Based Ad-Hoc Networks," *Int. J. Comput. Networks Appl.*, vol. 10, no. 4, pp. 668–687, 2023, doi: 10.22247/ijcna/2023/223319.

[53] P. Menakadevi and J. Ramkumar, "Robust Optimization Based Extreme Learning Machine for Sentiment Analysis in Big Data," in *2022 International Conference on Advanced Computing Technologies and Applications, ICACTA 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICACTA54488.2022.9753203.

[54] J. Ramkumar, R. Karthikeyan, and K. O. Nitish, "Securing Library Data With Blockchain Advantage," in *Enhancing Security and Regulations in Libraries with Blockchain Technology*, 2024, pp. 117–138. doi: 10.4018/979-8-3693-9616-2.ch006.

[55] K. S. J. Marseline, J. Ramkumar, and D. R. Medhunhashini, "Sophisticated Kalman Filtering-Based Neural Network for Analyzing Sentiments in Online Courses," in *Smart Innovation, Systems and Technologies*, A. K. Somani, A. Mundra, R. K. Gupta, S. Bhattacharya, and A. P. Mazumdar, Eds., Springer Science and Business Media Deutschland GmbH, 2024, pp. 345–358. doi: 10.1007/978-981-97-3690-4_26.

[56] P. S. Ponnukumar, N. I. Francis Xavier, and R. Jaganathan, "Stable Plithogenic Cubic Sets," *J. Fuzzy Ext. Appl.*, vol. 6, no. 2, pp. 410–423, 2025, doi: 10.22105/jfea.2025.449408.1422.

[57] J. Ramkumar and R. Vadivel, "Whale optimization routing protocol for minimizing energy consumption in cognitive radio wireless sensor network," *Int. J. Comput. Networks Appl.*, vol. 8, no. 4, pp. 455–464, 2021, doi: 10.22247/ijcna/2021/209711.

[58] S. R. Ziyad, Liyakathunisa, E. Aljohani, and I. A. Saeed, "Diagnosis of Autism Spectrum Disorder by Imperialistic Competitive Algorithm and Logistic Regression Classifier," *Comput. Mater. Contin.*, vol. 77, no. 2, pp. 1515–1534, 2023, doi: https://doi.org/10.32604/cmc.2023.040874.

[59] X. Gu, L. Xie, Y. Xia, Y. Cheng, L. Liu, and L. Tang, "Autism spectrum disorder diagnosis using the relational graph attention network," *Biomed. Signal Process. Control*, vol. 85, p. 105090, 2023, doi: https://doi.org/10.1016/j.bspc.2023.105090.

[60] H. Duan *et al.*, "Atypical Salient Regions Enhancement Network for visual saliency prediction of individuals with Autism Spectrum Disorder," *Signal Process. Image Commun.*, vol. 115, p. 116968, 2023, doi: https://doi.org/10.1016/j.image.2023.116968.

[61] M. Mishra and U. C. Pati, "A classification framework for Autism Spectrum Disorder detection using sMRI: Optimizer based ensemble of deep convolution neural network with on-the-fly data augmentation," *Biomed. Signal Process. Control*, vol. 84, p. 104686, 2023, doi: 10.1016/j.bspc.2023.104686.

[62] A. I. Alutaibi, S. K. Sharma, and A. R. Khan, "Capsule DenseNet++: Enhanced autism detection framework with deep learning and reinforcement learning-based lifestyle recommendation," *Comput. Biol. Med.*, vol. 190, p. 110038, 2025, doi: https://doi.org/10.1016/j.compbiomed.2025.110038.

[63] A. Lakhan, M. A. Mohammed, K. H. Abdulkareem, H. Hamouda, and S. Alyahya, "Autism Spectrum Disorder detection framework for children based on federated learning integrated CNN-LSTM," *Comput. Biol. Med.*, vol. 166, p. 107539, 2023, doi: 10.1016/j.compbiomed.2023.107539.

[64] L. Xiao, S. Zhang, S. Wu, H. Huang, and Z. Hu, "Classification of autism spectrum disorders based on the adaptive fuzzy reasoning system," *Biomed. Signal Process. Control*, vol. 104, p. 107513, 2025, doi: https://doi.org/10.1016/j.bspc.2025.107513.

[65] B. Divya, N. Udayakumar, R. Yuvaraj, and A. Kavitha, "Classification of low-functioning and high-functioning autism using task-based EEG signals," *Biomed. Signal Process. Control*, vol. 85, p. 105074, 2023, doi: https://doi.org/10.1016/j.bspc.2023.105074.

[66] Daniel, N. Dominic, T. W. Cenggoro, and B. Pardamean, "Machine Learning Approaches in Detecting Autism Spectrum Disorder," *Procedia Comput. Sci.*, vol. 227, pp. 1070–1076, 2023, doi: 10.1016/j.procs.2023.10.617.

[67] A. J. R. and R. Senthilkumar, "DSVTN-ASD: Detection of Stereotypical Behaviors in Individuals with Autism Spectrum Disorder using a Dual Self-Supervised Video Transformer Network," *Neurocomputing*, vol. 624, p. 129397, 2025, doi: https://doi.org/10.1016/j.neucom.2025.129397.

[68] A. Pyszkowska, T. Gąsior, F. Stefanek, and B. Więzik, "Determinants of escapism in adult video gamers with autism spectrum conditions: The role of affect, autistic burnout, and gaming

motivation," *Comput. Human Behav.*, vol. 141, p. 107618, 2023, doi: https://doi.org/10.1016/j.chb.2022.107618.

[69] Y. Huang, Y. Zhang, M. Chen, X. Han, and Z. Pan, "STL Net: A spatio-temporal multi-task learning network for Autism spectrum disorder identification," *Biomed. Signal Process. Control*, vol. 106, p. 107678, 2025, doi: https://doi.org/10.1016/j.bspc.2025.107678.

[70] U. Jabbar *et al.*, "Machine Learning–Based Approach for Early Screening of Autism Spectrum Disorders," *Appl. Comput. Intell. Soft Comput.*, vol. 2025, no. 1, p. 9975499, Jan. 2025, doi: https://doi.org/10.1155/acis/9975499.

[71] D. P. Kavadi, V. R. R. Chirra, P. R. Kumar, S. B. Veesam, S. Yeruva, and L. K. Pappala, "A Hybrid Machine Learning Model for Accurate Autism Diagnosis," *IEEE Access*, vol. 12, no. November, pp. 194911–194921, 2024, doi: 10.1109/ACCESS.2024.3520009.