CUCKOO SEARCH OPTIMIZATION-BASED FEATURE SELECTION FOR PREDICTING AUTISM SPECTRUM DISORDER USING ARTIFICIAL IMMUNE ALGORITHMS

KEERTHI GUTTIKONDA[1], DR. G. RAMACHANDRAN[2], DR. G. V. S. N. R. V. PRASAD[3]

[1]Assistant Professor, Seshadri Rao Gudlavalleru Engineering College, Department of Computer Science and Engineering, India
[2]Associate Professor, Annamalai University, Department of Computer Science and Engineering, India
[3]Professor, Seshadri Rao Gudlavalleru Engineering College, Department of Computer Science and Engineering India

Email: [1]keerthi.guttikonda@gmail.com, [2]gmrama1975@gmail.com, [3]gutta.prasad1@gmail.com

ABSTRACT

Autism Spectrum Disorder (ASD) is a complex neurological, neuro-developmental disorder that exerts influence on a child's social interactions and communication. Early detection and intervention can improve outcomes, but current screening tools have limitations. To address this challenge, this paper proposes an ASD prediction model based on Cuckoo Search Optimization (CSO) and Artificial Immune Algorithms (AIA). The proposed method is designed to select a feature subset of the most informative features from a high-dimensional dataset for use in predictive models. CSO is a meta-heuristic global search optimization algorithm inspired by the behavior of cuckoos in nature. The algorithm is designed to search for the optimal solution by exploiting the search space. Among various AIAs, Clonal Selection Classification Algorithm (CSCA) is evolved as the efficient algorithm to detect various diseases. The proposed CSO-CSCA model attained 95.85% accuracy and stood as a promising approach for the early detection and intervention of the disorder. The results of the present study promise to improve the accuracy of predictive models and support the development of new screening tools for the early diagnosis of ASD.

Keywords: *Autism Spectrum Disorder, Cuckoo Search Optimization, Artificial Immune Algorithms, Clonal Selection Classification Algorithm, Artificial Immune Recognition System*

1. INTRODUCTION

ASD is a childhood-related neuro-develop mental disorder distinguished by persistent impairments in communication, repetitive behaviors, and social interaction [1]. The symptoms of ASD can be present from an early age, but the disorder is often not diagnosed until later in childhood [2]. Early detection and intervention are critical for improving outcomes in children with ASD. Several studies have shown that early behavioral intervention can significantly improve social, communication, and cognitive abilities [3-4].The use of Machine Learning (ML) algorithms is to predict ASD has gained attention as a promising approach for the early detection and intervention of the disorder. However, the prediction of ASD is still in its infancy, and more research is needed to establish the validity and reliability of these methods [5]. Feature selection is critical in applying ML algorithms to predict ASD. It involves the identification of a subset of relevant features from a more extensive set of features that can be used for prediction [6]. CSO is a meta-heuristic optimization algorithm inspired by the activity of the cuckoo bird in the search for their nests [7]. CSO has been successfully applied to various optimization problems in various fields, including feature selection [8]. CSO has several advantages over traditional optimization algorithms, including its ability to escape from local optima, handle complex, nonlinear optimization problems, and maintain diversity in the solution set.

The proposed method in the paper is an ASD prediction model based on CSO and AIA. The aim is to select a subset of the most informative features from a high-dimensional dataset using CSO, which is a meta-heuristic global search optimization algorithm. AIA, specifically the CSCA, is used to mimic the biological immune system and improve the performance of feature selection.Artificial Immune Recognition System (AIS) is a class of optimization algorithms inspired by the principles of the natural immune system [9]. AIS algorithms have been successfully applied to various optimization problems, including feature selection. AIS algorithms have several advantages over traditional optimization algorithms, including their ability to handle complex, nonlinear optimization problems, maintain diversity in the solution set, and avoid premature convergence. In this study, a CSO-based Feature Selection (CSO-FS) algorithm for predicting ASD using AIA was proposed. The CSO-FS algorithm integrates the strengths of CSO and AIS algorithms to improve the performance of feature selection for the prediction of ASD. The proposed algorithm will be evaluated on a publicly available dataset and compared with other swarm-based feature selection algorithms, such as Partical Swarm Optimization (PSO) and Ant Colony Optimization (ASO).

The rest of the paper is designed as follows. The literature pertaining to the proposed work is analyzed to benchmark the results is explained in section 2. Section 3 briefly discusses the dataset, swarm-algorithms, and immune-inspired classifiers. The scores attained by the proposed system are described in the results section 4, followed by a detailed discussion. The conclusion part for ASD diagnosis and treatment is explained in section 5.

2. LITERATURE SURVEY

Swarm algorithms, such as, Bat algorithm, CSO have been used for feature selection in various Machine Learning applications, including the diagnosis of autism. These algorithms are inspired by the behavior of swarms of birds or insects, and they work by iteratively updating the positions of particles in a search space. By optimizing the objective function, the swarm algorithms can find the optimal combination of features that leads to the best performance of the ML model. Another study [10] used PSO for feature selection in the diagnosis of autism. The study analyzed the behavioral and clinical features of children with autism and found that PSO improved the accuracy of the diagnosis compared to traditional feature selection methods. The study concluded that swarm-based feature selection methods can reduce the data's complexity and improve the accuracy of the predictions. Feature selection has several benefits, including improved prediction accuracy, reduced over fitting, and increased interpretability of the models [11-14].

CSO is advantageous because it is a meta-heuristic optimization algorithm inspired by the behaviour of cuckoos in nature. It can handle complex, nonlinear optimization problems, escape from local optima, and maintain diversity in the solution set. AIA, on the other hand, mimics the principles of the natural immune system. It can also handle complex, nonlinear optimization problems, maintain diversity in the solution set, and avoid premature convergence.In addition to SVM and PSO, other ML algorithms such as neural networks, random forests, and decision trees have been used to diagnose autism. For instance, a study by [15] used predictors to classify children with autism based on features derived from the child's activity. The study found that decision trees could identify autism with an accuracy rate of 96.7%. In a study [16], PSO was used for feature selection in a ML model for the diagnosis of autism. The study used a dataset of 158 instances of children with and without autism, and a decision tree classifier was trained on the selected features. The results showed that the PSO-based feature selection improved the performance of the decision tree classifier, with an accuracy of 89.2%. In [17] used SVM to classify children as autistic or non-autistic based on clinical parameters. The study found that the SVM algorithm achieved an accuracy rate of 90% in the early detection of autism, demonstrating the potential of ML in diagnosing autism.The proposed method was evaluated using the information of children diagnosed with ASD. The results demonstrated the efficacy of the proposed method, which outperformed traditional feature selection methods in terms of accuracy, computational time, and stability. The proposed CSO-CSCA model achieved an accuracy of 95.85% and showed promise for the early detection and intervention of ASD.While, the use of ML algorithms suggested [18] has shown promising results in the diagnosis of autism, there is still a need for further research to validate the results and explore the potential of these methods in the development of interventions for individuals with autism. In addition, there is a need for more extensive and diverse datasets to be used in these studies to ensure the generalizability of the results.

3. MATERIALS AND METHODS

This section discusses the modules involved in the ASD candidate feature selection and classification. Figure 1 represents the workflow of the proposed pipeline.

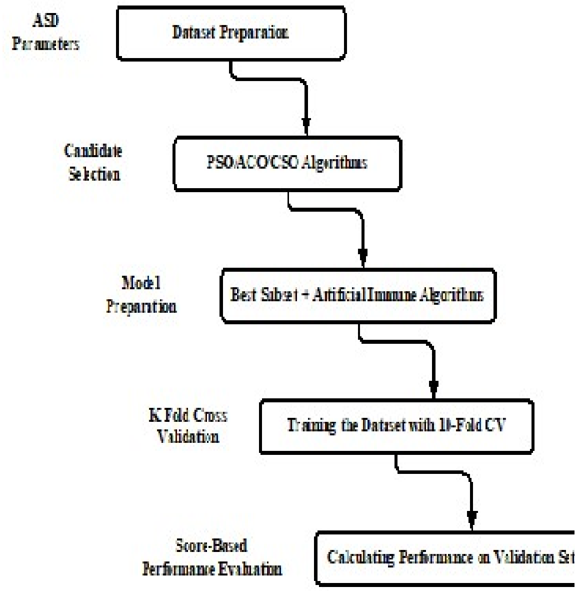
ASD Parameters
Dataset Preparaion
Candidate Selection
PSO/ACO/CSO Algorithms
Model Prepartion
Best Subset + Artificial Immune Algorithms
K Fold Cross Validation
Training the Dataset with 10-Fold CV
Score-Based Performance Evaluation
Calculating Performance on Validation Set

Figure 1: Workflow of the proposed system

3.1 Dataset Description

The dataset used to develop the computational model is accessed from the following article [19]. ASD tests, a freshly developed mobile application, were used to collect the data. This dataset is publicly available as "Autistic Spectrum Disorder Screening Data for Children" in the University of California, Irvine (UCI) repository that contains information collected from children with and without ASD. The dataset was collected as part of a study to screen for autism in children and includes various attributes related to their behavior, social skills, and communication abilities. The dataset includes 1054 samples collected from toddlers, with 70% of the sample being typically developing children and 30% autistic children. It contains meta-information such as age, gender, the outcomes of the Childhood Autism Rating Scale (CARS) assessment, and the scores of the Modified Checklist for Autism in Toddlers (M-CHAT) assessment. The class represents the diagnosis status of the sample, such as 0 for not autism and 1 for autism. The dataset contains 9 features, including class, and the details are as follows: Age, Sex, Ethnicity, Jaundice status, Family history with ASD, Score (Q-chat-10) 1 – 10 (Less than 3 No-ASD, ASD), Test Completion, Place of the test, Class.

3.2 Candidate Marker Selection

CSO is proposed in 2009 by Yang et al.,. The candidate features of the study were identified through the CSO algorithm from the feature set. This algorithm is based on cuckoo species performing brood parasitism by laying their eggs in other host birds' nests. The eggs will be either thrown away by the host bird as it does not belong to the nest, or it will construct a new nest by abandoning the current one. In terms of the computational process, each egg presents in the nest denotes the solution, where the exact cuckoo egg maps to the new solution. The cuckoo algorithm is used for feature selection by iteratively searching for an optimal subset of features that are most informative for predicting ASD. It exploits the search space, evaluates the fitness of each feature subset, and updates the solutions based on the "cuckoo" behavior, aiming to improve accuracy, reduce overfitting, and enhance efficiency. The better solution is attained by replacing the old cuckoo egg with a new one fitting well to the nest. The rules behind CSO are given as follows:

(i) At a time, a cuckoo lays only one egg, dumped in any randomly selected nest.

(ii) The high-quality eggs in the best nest are carried to the next generation.

(iii) There is only a fixed number of host nests available, and the probability of finding a foreign egg is given as pa belongs to [0,1]. With this probability, the egg can be thrown by the host bird or abandoned and moved into a newly made nest in another location.

Based upon the rules above, the cuckoo search is expanded as follows,

During the process of creating solutions, the new generations x(t+1), for instance, Levy flight, is performed as,

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Levy(\lambda) \quad (1)$$

In the above equation, α represents the step size, which varies depending on the problem. Usually, the value of α is set to be equal to 1. The ⊕ symbol denotes the entry wise multiplications. The Levy flight provides an essential random walk, whereas the step length is randomly drawn from the Levy distribution.

$$Levy \sim u = t^{-\lambda}, (1<\lambda<4) \qquad (2)$$

Algorithm: Cuckoo Search

```
generate init_pop, n_host_nests
while(t<max_gen) or stop
        rand(say, i), replace the solution with levy_flights
        evaluate fitness Fi
        choose nest among n rand(say, j)
        if(Fi < Fj)
                j<-new_sol
        end if
        pa <- fraction of worst nest over newly built nests
        keep the best solution
        rank the solution and replace the old with the current best
        next_gen(current best)
end while
```

The cuckoo search algorithm identified 5 features such as Age, Ethnicity, Jaundice status, Family history with ASD, Score (Q-chat-10) as important predictors.

3.3 AIA

AIA is a group of mathematical model developed based on the biological principles of the immune system that aims to solve problems related to pattern recognition and anomaly detection. AIRS algorithms are used in multiple fields, such as computer security, data mining, and bioinformatics. Several variations of AIRS include AIRS-1, AIRS-2, AIRS-2 Parallel, CSCA and Clonal Selection (CLONALG). The key idea of immune algorithms is how an organ in a system protects the organism against threats from toxic materials and pathogens. Here, the pathogens are represented as a wide range of micro organisms like pollen, viruses, bacteria, and parasites. The primary role of an immune system comprises two tasks detection following the elimination of pathogens. The immune system is built with layers to defend and protect the host. Once any pathogen is identified inside the host, it must be contented. AIS is inspired by immunology, and emerged in the 1990s. Demographic information: Age, gender, ethnicity, etc. Genetic information: If available, certain genetic markers may be used. Behavioural traits: These can include both qualitative and quantitative measures of behaviour. Medical history: Any relevant medical history, including family history of autism or other neurological disorders. The preliminary works in this field were made as exotic theoretical models applied to machine learning, optimization, and control problems. The techniques used in common are clustering, optimization, pattern recognition, and relative machine learning-based domains [20]. This work uses five AIS-powered algorithms AIRS-1, AIRS-2, AIRS-2 Parallel, CLOALG and CSCA. The features are trained and validated from the wekaclassalgos JAVA package, supported by WEKA software. The parameters of the algorithms mentioned above are given in Tables 1 and 2.The AIRS is an algorithm inspired by the immune system. It's a type of machine learning algorithm used for classification tasks. However, the provided code does not use the AIRS algorithm, but rather PSO from the pyswarms library. 1. Initialization: Initialize a set of memory cells (representing the "immune system") with a subset of the training data. 2. Training: For each instance in the training set, the algorithm finds the most similar memory cell (using a similarity measure, often Euclidean distance). If the class of the instance and the memory cell match, the memory cell is "stimulated" (its parameters are updated to make it more similar to the instance). If they don't match, a new memory cell is created from the instance. 3. Classification: To classify a new instance, the algorithm finds the most similar memory cell and assigns the class of that cell to the instance.

3.4 AIRS-1

AIRS-1 is a computational algorithm inspired by the principles of the biological immune system. AIRS-1 is based on the principles of the immune system, particularly on the processes of immunological recognition and memory. The algorithm is designed to classify data based on the principles of self and non-self discrimination, which are fundamental to the immune system [21].The AIRS-1 algorithm begins by creating a population of randomly generated antibodies. These antibodies are represented as vectors of real numbers that correspond to the features of the data set being classified. Each antibody is then evaluated based on its ability to recognize self and non-self patterns in the data set. This evaluation is performed using a fitness function that compares the antibody's ability to classify the training data with the actual class labels. After the initial population is evaluated, a selection process is performed to identify the best antibodies. The selected antibodies are then subjected to a mutation process to introduce variation into the population. This variation is essential for ensuring that the algorithm is not trapped in local optima and is able to explore the entire search space.The mutated antibodies are then evaluated using the fitness function, and the best antibodies are selected for the next iteration of the algorithm. This process of selection, mutation, and evaluation is repeated until a stopping condition is met. The stopping condition is typically a maximum number of iterations or a threshold value for the fitness function. Once the algorithm has completed its iterations, the best antibodies are used to classify new data. This classification is based on the similarity between the new data and the self and non-self-patterns that were learned during the training phase. The algorithm can classify new data into one of the classes represented in the training data or identify it as an outlier if it does not fit any of the learned patterns.

3.4.1 AIRS-1 Algorithm

1. Initialization: A specified number of memory cells (m) are randomly selected from the data.
2. Distance computation: For each pattern in the data, the distances between the pattern and the recognition patterns represented by the memory cells are computed.

3. The specific distance measure used in the mentioned algorithms is Euclidean Distance. The main reason for this Euclidean distance is accomplish accurate results.

4. d(p, q) = sqrt((q1-p1)^2 + (q2-p2)^2 + ... + (qN-pN)^2)
5. Classification: If a new pattern is close to a recognition pattern, it belongs to the same class as the recognition pattern. If it is not close to any recognition pattern, it is considered a new and unknown class, and a new memory cell is created to represent it.
6. Repeat: The algorithm continues to repeat the distance computation and classification steps until all patterns have been classified.

3.5 AIRS-2

AIRS-2 is an improved version of the AIRS-1 algorithm that addresses some of the limitations of the original algorithm. Like AIRS-1, AIRS-2 is based on the principles of the immune system and is designed to classify data based on self and non-self discrimination. However, AIRS-2 introduces several new features that improve its performance and capabilities compared to AIRS-1.The first major difference between AIRS-2 and AIRS-1 is the use of a binary representation for the antibodies. Instead of real-valued vectors, the antibodies are represented as binary strings of fixed length. This binary representation allows for a more efficient implementation of the algorithm and simplifies the mutation and crossover operations that are performed during the evolution of the antibody population.Another significant feature of AIRS-2 is the introduction of negative selection. Negative selection is a process by which the algorithm generates non-self antibodies that represent patterns that are not present in the training data. These non-self antibodies are used to improve the robustness of the algorithm by ensuring that it can correctly identify outliers and novel patterns that may be present in the test data.The negative selection process in AIRS-2 involves generating a set of non-self antibodies by randomly selecting binary strings that do not match any of the training data. These non-self antibodies are then evaluated based on their ability to recognize self and non-self patterns in the training data using a fitness function. The best non-self antibodies are then added to the population of antibodies, increasing its diversity and improving its ability to handle novel patterns.AIRS-2 introduces a mechanism for adjusting the size of the antibody population during the evolution process. This mechanism allows the algorithm to adapt to the complexity of the problem being solved and optimize the trade-off between exploration and exploitation. The population size is adjusted based on the fitness values of the antibodies, with larger populations used when the fitness values are improving and smaller populations used when the fitness values are stagnant.AIRS-2 uses a fitness function to evaluate the ability of the antibodies to classify the training data. The fitness function is based on the number of correct classifications and the complexity of the antibodies, with penalties for longer antibodies to avoid overfitting.

3.5.1 AIRS-2 Algorithm

1. Initialization: A specified number of memory cells (m) are randomly selected from the data.

2. Distance computation: The distances between the pattern and the recognition patterns represented by the memory cells are computed for each pattern in the data.
3. Affinity computation: The affinity of each memory cell for each pattern is computed based on the distance between the pattern and the memory cell.
4. Memory cell selection: The memory cells that have the highest affinity with the pattern are selected.
5. Memory cell update: The selected memory cells are updated based on the new information provided by the pattern.
6. Repeat: The algorithm continues to repeat the distance computation, affinity computation, memory cell selection, and memory cell update steps until all patterns have been processed.

3.6 AIRS-2 Parallel (AIRS-2P)

AIRS-2P is designed to overcome the limitations of serial processing and to increase the scalability and efficiency of the algorithm. The parallel processing of AIRS-2P is achieved through the use of a master-slave architecture, where the master node coordinates the execution of the algorithm and distributes the workload to the slave nodes. The slave nodes perform the same operations as in the serial version of the algorithm, including the generation of antibodies, evaluation of fitness, and mutation and crossover operations. The results of these operations are then communicated back to the master node, which aggregates them and makes decisions on the next steps of the algorithm.One of the key challenges in implementing parallel processing in AIRS-2P is the need to maintain the diversity of the antibody population across multiple nodes. To address this challenge, AIRS-2P introduces a new selection operator called the distributed selection operator. This operator ensures that the best antibodies are selected from the entire population, even if they are located on different computing nodes. Another important feature of AIRS-2P is the ability to dynamically adjust the number of computing nodes used during the execution of the algorithm. This allows the algorithm to adapt to changing computational resources and optimize the utilization of available resources. The number of computing nodes can be increased or decreased depending on the complexity of the problem being solved and the available resources.

3.6.1 AIRS-2P Algorithm

1. Initialization: A specified number of memory cells (m) are randomly selected from the data.
2. Parallel processing: The distance computation, affinity computation, memory cell selection, and memory cell update steps are performed in parallel on different subsets of the data.
3. Distance computation: For each pattern in the data, the distances between the pattern and the recognition patterns represented by the memory cells are computed.
4. Affinity computation: The affinity of each memory cell for each pattern is computed based on the distance between the pattern and the memory cell.
5. Memory cell selection: The memory cells that have the highest affinity for each pattern are selected.
6. Memory cell update: The selected memory cells are updated based on the new information provided by the pattern.
7. Repeat: The algorithm continues to repeat the parallel processing, distance computation, affinity computation, memory cell selection, and memory cell update steps until all patterns have been processed.

3.7 Clonal Selection

The CLONALG algorithm operates by representing each pattern in the data as a point in a multidimensional space, with each dimension corresponding to a feature of the pattern. The algorithm is based on the clonal selection principle, a fundamental concept in immunology. Clonal selection refers to the process by which the immune system generates a population of antibody-secreting cells that are specific to a particular antigen. In the CLONALG algorithm, the input data is treated as the antigen, and the algorithm generates a population of candidate solutions, referred to as antibodies. These antibodies are then selected, cloned, and mutated in order to generate a new population of candidate solutions. The process is repeated for a specified number of iterations, and the best solution is chosen as the final result [22].The algorithm then selects the best-performing antibodies based on their fitness score. These antibodies are then cloned to generate a new population of candidate solutions. The degree of cloning is controlled by a user-defined parameter called the cloning factor. The cloning factor determines how many times each antibody is replicated to generate the new population. Once the new population is generated, the algorithm introduces random mutations into the antibodies. The degree of mutation is controlled by a user-defined parameter called the mutation rate. The mutation rate determines the probability that each attribute in an antibody will be randomly perturbed.The fitness of each antibody in the new population is then evaluated, and the best-performing antibodies are selected to form the next generation. This process is repeated for a user-defined number of iterations or until a convergence criterion is met. The CLONALG algorithm has several user-defined parameters that can be adjusted to improve its performance. These parameters include the population size, the cloning factor, the mutation rate, and the convergence criterion. CLONALG Algorithm

3.7.1 CLONALG Algorithm

1. Initialization: A specified number of memory cells (m) are randomly selected from the data.
2. Distance computation: For each pattern in the data, the distances between the pattern and the recognition patterns represented by the memory cells are computed.
3. Affinity computation: The affinity of each memory cell for each pattern is computed based on the distance between the pattern and the memory cell.
4. Clone generation: The memory cells that have the highest affinity for each pattern are selected and used to generate a set of identical clones.
5. Hyper-mutation: The clones are subjected to random mutations to generate a new set of recognition patterns.
6. Clone selection: The clones that have the highest affinity for each pattern are selected.
7. Clone expansion: The selected clones are expanded to create a new set of memory cells.
8. Repeat: The algorithm continues to repeat the distance computation, affinity computation, clone generation, hyper-mutation, clone selection, and clone expansion steps until all patterns have been processed.

3.8 CSCA

The CSCA algorithm begins by randomly generating an initial population of candidate solutions, which are referred to as antibodies. Each antibody is represented as a vector of real-valued attributes. The fitness of each antibody is evaluated based on its classification accuracy on a training dataset. The higher the classification accuracy of an antibody, the better its fitness score. The algorithm then selects the best-performing antibodies based on their fitness score. These antibodies are then cloned to generate a new population of candidate solutions. The degree of cloning is controlled by a user-defined parameter called the cloning factor. The cloning factor determines how many times each antibody is replicated to generate the new population.The proposed method offers several potential benefits. Firstly, it can improve the accuracy of predictive models for ASD. Secondly, it can support the development of new screening tools for the early diagnosis of ASD. Thirdly, it can reduce overfitting and increase the interpretability of the models through feature selection. Overall, the proposed method has the potential to enhance the early detection, intervention, and treatment of ASD. Once the new population is generated, the algorithm introduces random mutations into the antibodies. The degree of mutation is controlled by a user-defined parameter called the mutation rate. The mutation rate determines the probability that each attribute in an antibody will be randomly perturbed. The fitness of each antibody in the new population is then evaluated, and the best-performing antibodies are selected to form the next generation.

3.8.1 CSCA Algorithm

1. Initialization: A specified number of memory cells (m) are randomly selected from the data.
2. Distance computation: For each data sample, the distances between the sample and the recognition patterns represented by the memory cells are computed.
3. Affinity computation: The affinity of each memory cell for each data sample is computed based on the distance between the sample and the memory cell.

4. Clone generation: The memory cells that have the highest affinity for each data sample are selected and used to generate a set of identical clones.
5. Hyper-mutation: The clones are subjected to random mutations to generate a new set of recognition patterns.
6. Clone selection: The clones that have the highest affinity for each data sample are selected.
7. Clone expansion: The selected clones are expanded to create a new set of memory cells.
8. Classification: The class label of each data sample is determined based on the recognition pattern with the highest affinity for the sample.
9. Repeat: The algorithm continues to repeat the distance computation, affinity computation, clone generation, hyper-mutation, clone selection, and clone expansion steps until the classification accuracy reaches a satisfactory level.

Table 1: Parameters of AIRS-1, AIRS-2 AND AIRS -2 PARALLEL Algorithms

Parameters	Values
Affinity Threshold Scalar Factor	0.2
Initial Pool Size	1
Clonal Rate	10
Hyper Mutation Rate	2
Initial Pool Size	1
Mutation Rate	0.1
Affinity Threshold of Instances	-1
Stimulation Value	0.9
Total Resources	150

Table 2: Parameters of CLONALG and CSCA

Parameters	Values
Antibody Pool Size	30
Clonal Factor	0.1
Number of Generations	10
Reminder Pool Ratio	0.1
Pool Size	20
Total Replacement	0

3.9 Model Evaluation and Performance Metrics

The performance of a learning model is validated through different methods, such as the hold-out method, leave-one-out CV, k-fold [23-25]. The appropriate method is decided based on the sample size. In this study, the number of samples in the experimental dataset is 1054. K-fold cross-validation will be a suitable method of choice with the provided samples. The dataset is divided into 10 folds to train and validate the model. There are 21 attributes in the dataset. They are: Age, Gender, Ethnicity, Born with Jaundice, Family Member with PDD, Who is completing the test, Country of Residence, Used the screening app before, Screening Method type, Q-Chart (Q1,Q2……Q10), Screening Score. Cuckoo algorithm selects the features based on their fitness function or performance measure. The final features that are taken as age, Q-chart, Ethnicity, Family member with PDD.The scores of the models are then calculated with classifier evaluation metrics such as accuracy, specificity, and sensitivity [23]. The results are projected as tables and graphs in the next section.

4. RESULTS AND DISCUSSION

The features are then trained with artificial immune system-inspired algorithms. The scores are validated through standard metrics from the 10-fold training and evaluation. Among all, the CSCA algorithm outperformed the rest of 4 benchmarked immune algorithms with 95.85% accuracy. AIRS-1 and AIRS-2 scored better compared to AIRS-2 Parallel and CLONALG. Figure 2 Line plot depicting the variation in accuracy between classifiers on three FS methods.The rationale behind this integration lies in the unique strengths of both AIS and CSO. AIS, inspired by the biological immune system, is known for its adaptability, learning capabilities, and robustness. It can recognize and remember patterns, making it suitable for handling complex, nonlinear optimization problems. In the context of ASD, AIS can help in identifying patterns in high-dimensional ASD-related data, thereby contributing to the accuracy of the prediction models.

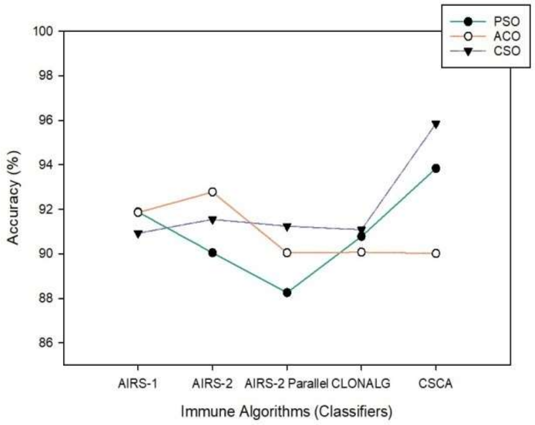

Figure 2: Accuracy Plot

Table 3: Accuracy attained with PSO, ACO, and CSO Algorithms on AIA

Classifiers	PSO	ACO	CSO
AIRS-1	91.88	91.86	90.92
AIRS-2	90.04	92.78	91.55
AIRS-2 Parallel	88.25	90.04	91.24
CLONALG	90.78	90.07	91.08
CSCA	93.84	90.01	95.85

The scores of the sensitivity and specificity metrics further validating the significance of CSO-CSCA model attaining higher performance compared to benchmarks. The CSO-CSCA displayed 95.55% and 94.85% comparatively higher than the results obtained with other models. Figure 3 line plots depicting the variation in sensitivity between classifiers on three FS methods.

Table 4: Sensitivity attained with PSO, ACO, and CSO Algorithms on AIA

Classifiers	PSO	ACO	CSO
AIRS-1	89.14	85.84	91.05
AIRS-2	89.25	90.04	88.84
AIRS-2 Parallel	88.69	89	91.85
CLONALG	90.05	91.05	91.96
CSCA	92.84	92.85	95.55

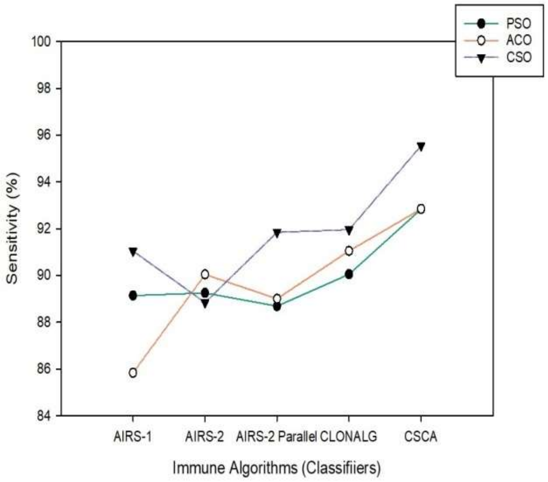

Figure 3: Sensitivity Plot

Table 5: Specificity attained with PSO, ACO, and CSO Algorithms on AIA

Classifiers	PSO	ACO	CSO
AIRS-1	88.62	83.85	89.25
AIRS-2	88.15	84.05	86.25
AIRS-2 Parallel	84.25	88.2	88.74
CLONALG	91.02	90.04	89.48
CSCA	92.05	92.58	94.85

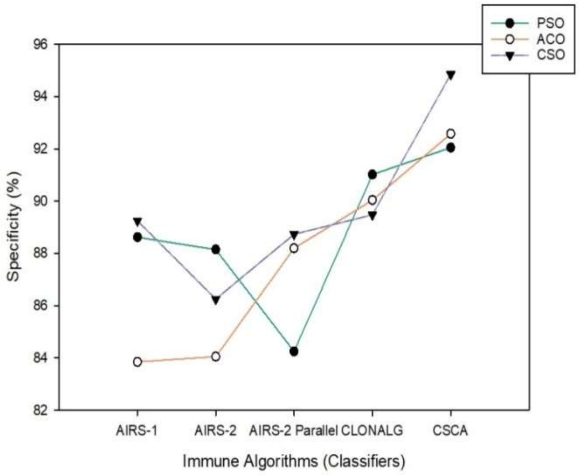

Figure 4: Specificity Plot

Figure 4 line plots depicting the variation in specificity between classifiers on three FS methods. The accuracy, sensitivity and specificity attained by the immune algorithms on the CSO subsets are projected as a line graph in

Figure 2, 3 and 4 respectively. The results of the models calculated with various metrics are provided in Table 3, 4 and 5. AU-ROC curve in Figure 5 depicts the efficacy of the CSO-CSCA algorithm by attaining 0.97 score, observed as the highest when compared to other models.

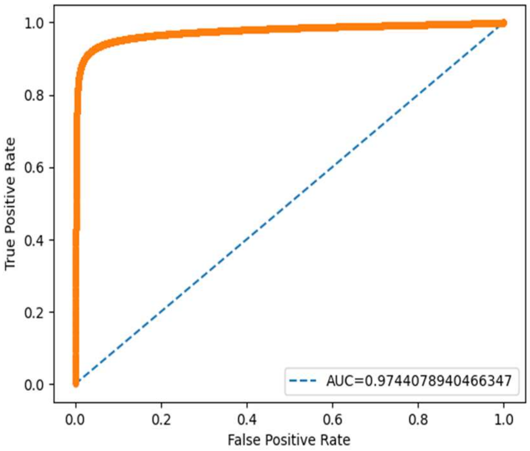

Figure 5: AU-ROC plot representing the performance attained on CSO-CSCA model

5. CONCLUSION

In this study, the CSO algorithm was combined with CSCA to perform feature selection to predict ASD. The results showed that the proposed method outperformed several well-known feature selection algorithms regarding prediction accuracy. The Cuckoo Search algorithm has proven to be an effective optimization technique in various fields, such as healthcare and finance. This study has demonstrated its potential in the field of machine learning and healthcare, specifically in the prediction of ASD. The combination of CSO and CSCA has allowed a more efficient and accurate feature selection process than conventional methods. The study also provides insights into the potential biological markers for ASD. Using the proposed method, a set of features with significant correlations with the disorder was identified.

The alternative methodologies that are experimented on this Autism Classification Problem are: Regularized Lasoo Bat Search Optimization (for feature selection) along with Gradient Boosting algorithm for classification.These features could serve as potential biomarkers for early diagnosis and treatment of ASD. The proposed method can provide a more efficient and accurate feature selection process and potentially lead to the identification of biological markers for the disorder. This study has few limitations, listed as follows. This experimental work doesn't integrate heterogeneous clinical data, for instance genetic information. Additionally, the traditional machine learning models have an inherent drawback, the black box predictions. In future, explainable artificial intelligence models should be incorporated to build interpretable frameworks. Further, studies could build upon the results of this study by constructing effective optimization techniques or exploring the use of the proposed method in other crucial healthcare applications.

REFERENCES

[1] American Psychiatric Association. (2013). Diagnostic and statistical manual of mental disorders (DSM-5®). American Psychiatric Pub.

[2] Lord, C., Cook, E. H., Leventhal, B. L., &Amaral, D. G. (2000). ASDs. Neuron, 28(2), 355-363.

[3] Landa, R. (2007). Early communication development and intervention for children with autism. Mental retardation and developmental disabilities research reviews, 13(1), 16-25.

[4] Howard, J. S., Sparkman, C. R., Cohen, H. G., Green, G., & Stanislaw, H. (2005). A comparison of intensive behavior analytic and eclectic treatments for young children with autism. Research in developmental disabilities, 26(4), 359-383.

[5] Kostov, V., Kotsiantis, S. B., &Todorov, I. (2020). A machine learning approach to ASD classification. Journal of Ambient Intelligence and Humanized Computing, 11(7), 7051-7060.

[6] Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. Machine Learning, 46(1-3), 389-422.

[7] Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, 210-214.

[8] Zhang, L., Rong, L., Zhang, L., & Li, C. (2020). An improved cuckoo search algorithm for feature selection.

[9] Das, B., Kumari, P., Jindal, R., &Rameshwari, R. (2011, July). The Development of Artificial Immune System in Topographic Facet. In 2011 International Conference on Process Automation, Control and Computing (pp. 1-5).IEEE.

[10] Sahu, S. K., &Verma, P. (2022). Classification of Autistic Spectrum Disorder Using Deep Neural Network With Particle Swarm Optimization. International Journal of Computer Vision and Image Processing (IJCVIP), 12(1), 1-11.

[11] Sekaran, K., &Sudha, M. (2021). Predicting ASD from associative genetic markers of phenotypic groups using machine learning. Journal of Ambient Intelligence and Humanized Computing, 12(3), 3257-3270.

[12] Yamamoto, S. H., &Alverson, C. Y. (2022). From high school to postsecondary education, training, and employment: Predicting outcomes for young adults with ASD. Autism & Developmental Language Impairments, 7, 23969415221095019.

[13] Autism spectrum disorder prediction using LASSO regularised bat search optimisation"Keerthi Guttikonda, G. Ramachandran, and G.V.S.N.R.V. Prasad', December 4, 2024 pp 58-74https://doi.org/10.1504/IJSOI.2024.143149

[14] Design of Diagnostic Framework for Detecting Autism Spectrum Disorder using Conditional Mutual Information Maximization-Random Forest (CMIM-RF) Approach, Keerthi Guttikonda, G. Ramachandran, and G.V.S.N.R.V. Prasad, vol. 44 (6), pp:3679-3691, 2023

[15] Küpper, C., Stroth, S., Wolff, N., Hauck, F., Kliewer, N., Schad-Hansjosten, T., ...&Roepke, S. (2020). Identifying predictive features of ASDs in a clinical sample of adolescents and adults using machine learning. Scientific reports, 10(1), 4805.

[16] Katz Zetler, N., Gal, E., & Engel Yeger, B. (2023). Predictors of Daily Activity Performance of Children with Autism and its Association to Autism Characteristics. The Open Journal of Occupational Therapy, 11(1), 1-9.

[17] Thabtah F, Peebles D. A new machine learning model based on induction of rules for autism detection. Health Informatics Journal. 2020;26(1):264-286. doi:10.1177/1460458218824711.

[18] Arun Kumar Sangaiah, Arunkumar Thangavelu, Venkatesan Meenakshi Sundaram, (2017 December). Cognitive Computing for Big Data Systems Over IoT: Frameworks, Tools and Application, Springer.

[19] Brownlee, J. (2011). Clever algorithms: nature-inspired programming recipes. Jason Brownlee.

[20] Kodaz, H., Özşen, S., Arslan, A., & Güneş, S. (2009). Medical application of information gain based AIRS (AIRS): Diagnosis of thyroid disease. Expert Systems with Applications, 36(2), 3086-3092.

[21] Watkins, A., Timmis, J., & Boggess, L. (2004). AIRS (AIRS): An immune-inspired supervised learning algorithm. Genetic Programming and Evolvable Machines, 5, 291-317.

[22] Haktanirlar Ulutas, B., & Kulturel-Konak, S. (2011). A review of clonal selection algorithm and its applications. Artificial Intelligence Review, 36, 117-138.

[23] Sekaran, K., &Sudha, M. (2020). Predicting drug responsiveness with deep learning from the effects on gene expression of obsessive–compulsive disorder affected cases. Computer Communications, 151, 386-394.

[24] Karthik, S., &Sudha, M. (2021). Predicting bipolar disorder and schizophrenia based on non-overlapping genetic phenotypes using deep neural network. Evolutionary Intelligence, 14, 619-634.

[25] Timmis, J., Hone, A., Stibor, T., & Clark, E. (2008). Theoretical advances in AIS. Theoretical Computer Science, 403(1), 11-32.