[1]Sudipta Priyadarshinee

[2]Madhumita Panda

Enhancing Diabetes Prediction and Classification through Metaheuristic Optimization of Deep Neural Networks

Journal of Electrical Systems

Abstract: - Deep Neural Network (DNN) classifiers have found numerous applications in health care due to their ability to learn complex patterns and extract meaningful insights from medical data. These models have parameters that are not learned from the data but need to be set before the training process begins. The choice of hyperparameters can significantly impact the performance of a model. Fine-tuning these hyperparameters is a crucial step in optimizing a model for a specific task. Machine learning and metaheuristic approaches play vital roles in this process. This paper presents a hybrid metaheuristics algorithm using Coati Optimization and Cellular Automata to provide a systematic and automated approach in exploring the hyperparameter space of Deep Neural Network. To evaluate the model performance, the experiments were conducted with PIMA Diabetes dataset and a real-world data set collected from a local pathology clinic. Through stratified k-fold cross-validation, the proposed hybrid model obtains 95.73% accuracy on PIMA dataset while on the real dataset it achieves 99.9% accuracy. The strengths of our proposed model are further compared with other machine learning algorithms demonstrating its potential to revolutionize diabetes research.

Keywords: Metaheuristic Algorithms, Deep Neural Network, Classification, Diabetes Prediction

I. INTRODUCTION

Present days, Machine learning (ML) an offspring of artificial intelligence has been adopted across a wide range of industries, from finance, marketing to health sector. However, its impact in the health sector cannot be overstated. Healthcare is constantly changing due to the constant development of new technology and ideas. ML could assist medical professionals in some of these new scenarios. These algorithms help to find patterns and insights in medical data that would be impossible to find manually which further helps in better clinical decision support and the development of clinical practice guidelines within health systems. Nowadays, with the rapid growth of real-world problems and the importance of access speed to answers, classical methods cannot deal with many problems and random algorithms are mostly used. Therefore, in recent decades, the use of metaheuristic (MHs) algorithms has grown significantly. Moreover, the best performance from a machine learning model often requires setting its hyperparameters to optimal values and Metaheuristic techniques play a vital role in efficiently exploring the hyperparameter space to find the best configurations. These are optimization techniques that are designed to find high-quality solutions to optimization problems, especially in cases where the search space is large and complex. Hyperparameter tuning in machine learning can be framed as an optimization problem, where the goal is to find the set of hyperparameters that optimizes the performance of a model on a given task. Metaheuristic algorithms are particularly useful for this purpose due to their ability to efficiently explore complex solution spaces. Heat transfer search (HTS) [1], particle swarm optimization (PSO) [2], genetic algorithm (GA) [3], water cycle algorithm (WCA) [4], Monarch butterfly optimization (MBO) [5], Grey Wolf Algorithm (GWA) [6], lightning search algorithm (LSA) [7] and dragonfly algorithm (DA) [8] are a few examples of MHs algorithms.

In recent years, significant advancements have been made in the domain of MH algorithms, which serve many purposes, including feature selection and hyperparameter optimization. There are many types of algorithms based upon their inspiration such as PSO [2], Artificial Bee Colony (ABC) [16], Focus and Shake Algorithm (FSA) [47], and Swarm Bipolar Algorithm (SBA) [48] which are examples of swarm-based, GA [3] which is evolutionary-based, Carpet Weaver Optimization (CWO) [46], Sculptor Optimization Algorithm (SOA) [49], and Dollmaker

[1] School of Computer Science, G.M. University, Sambalpur, Odisha, India

[2]School of Computer Science, G.M. University, Sambalpur, Odisha, India [email:sudiptapatel88@gmail.com]

Optimization Algorithm (DOA) [50] are human-based, with Darts Game Optimizer (DGO) [51] and Golf Optimization Algorithm (GOA) [52] which are game-based methods.

In the medical field, MHs algorithms are employed to enhance classifier tuning owing to their exceptional global search capabilities for identifying optimal solutions. Nonetheless, most biomedical issues have high dimensionality, posing a challenge for optimization methods reliant on MHs because to the curse of dimensionality [61]. Researchers have deemed these algorithms as useful and reliable optimization procedures, especially for addressing complex, high-dimensional issues such as parameter tuning of Neural network classifiers. Recent MHs algorithms, including PSO [2], GA [3], FSA [47], and WOA [21], frequently encounter issues such as premature convergence, restricted scalability, and computing inefficiency in tasks with high-dimensional data. Recent MHs algorithms, including PSO [2], GA [3], FSA [47], CWO [46], SOA [49], DOA [50], SBA [48], and WOA [21], frequently encounter issues such as premature convergence, local optima, restricted scalability, and computing inefficiency in tasks with high-dimensional data. to address these challenges in the present work, the authors have utilized Coati Optimisation Algorithm (COA) to optimize DNN parameters. This COA addresses issues like strong balancing between exploration-exploitation, and computing efficiency. But similar to many other MHs, COA also suffers from premature convergence that prevents the algorithm from finding the best solution by trapping it in a local optimum. To avoid this trapping of local optima cellular automata (CA) is used. CA introduces a lattice structure for organizing candidate solutions. Each cell (i.e. solution) interacts with its neighbors, encouraging diversity while converging toward the global optimum. This lattice like architecture prevents premature convergence. CA further improves the COA by enhancing its ability to explore and exploit the search space more effectively. The proposed Coati Cellular Optimization (CCO) algorithm addresses these limitations by integrating the exploratory strengths of the COA with the neighbourhood enhancement features of CA.

This research work is broken up into seven phases, each of which advances knowledge and practise of machine learning algorithms for diabetes prediction. Section 2 offers the important research and studies that support the diabetes findings while Section 3 discusses deep neural network. Section 4 presents proposed hybridization technique, while Section 5 explains the evaluation of proposed hybridization. Section 6 discusses the simulation results as well as presents a comparison with previous scholarly works providing insights into the efficacy of the proposed approach. Finally, in Section 7, the work concludes with a complete review of the study's significant findings.

II. RELATED WORK

In the dynamic field of healthcare research, the convergence of metaheuristic optimisation methodologies with machine learning applications has resulted in a slew of novel studies, particularly in the diabetes and cardiovascular disease domains.

One notable contribution comes from [9], which uses data mining and metaheuristic methods to help diabetic patients avoid readmissions within 30 days. Using a dataset from 130 US hospitals, the study identifies 20 essential characteristics using Chi-square analysis, with GA-SVM achieving the highest accuracy (73.52%), demonstrating the efficacy of tailored approaches in demanding settings. The downside of GA is that, because it is an evolution-based algorithm, natural life tends to evolves away from bad situations instead towards positive ones [53]. Therefore, GA can lead to suboptimal optimizations.

Addressing the growing prevalence of type II diabetes, [10] presents hybrid methods for early detection. Using Particle Swarm Optimisation for feature selection and meta-heuristics for hyperparameter optimisation, the Genetic Algorithm-Support Vector Machine (GA-SVM) achieves an astonishing 93% accuracy. However, PSO frequently converges to local optima, particularly in multi-modal or complex search spaces, as a result of inadequate diversity among particles.

[11] investigates the importance of big data in diabetes prediction, emphasising the MayFly Algorithm (MA) (94% accuracy) and the Whale Optimisation Algorithm (WOA) (95.05% accuracy). The literature study goes into big data's importance in healthcare, optimisation algorithm applications, and current diabetes prediction studies. The Mayfly algorithm is limited by its dependence on initial parameter tuning, indicating that its performance is directly correlated with the selected parameter values. Similarly, the WOA also needs parameter tuning along with premature convergence rate [54].

The paper [12] focuses on early diabetes diagnosis by combining Harmony Search, Genetic Algorithm, and Particle Swarm Optimisation. The proposed method, which uses K-nearest neighbour for classification, has an accuracy of 91.65%, exceeding earlier approaches. Similar to the WOA Harmony search suffers from premature convergence [55].

AHDHS-Stacking, an ensemble learning system that classifies diabetes using stacking and the harmony search method, is introduced in this article [13]. The paper introduces an adaptive hyper-parameter adjustment strategy to help the algorithm jump out of the local optimum trap. High accuracy of 93.09% and 87.55% are obtained by the model when it is validated on the PID and CWMD datasets. With an emphasis on the novel contributions made by AHDHS-Stacking, the literature review will emphasise the value of early diabetes diagnosis and the applicability of machine learning in this field.

MF-CSA, a novel method for deep feature extraction in Convolutional Neural Networks (CNN) for diabetes detection and glucose level prediction, is described in this research [14]. By utilising the Crow Search Algorithm and Moth-Flame Optimisation to optimise feature selection, the MF-CSA algorithm enhances CNN performance. However, in this paper the Crow Search Algorithm and Moth-Flame Optimization are used for feature extraction, rather than optimizing the CNN, resulting in reduced classification accuracy.

This study [15] uses a Principal Component Analysis PCA-based Deep Neural Network (DNN) model optimised with the Grey Wolf Optimisation (GWO) algorithm to classify diabetic retinopathy features. The suggested model outperforms classic machine learning techniques such as SVM, Naive Bayes, Decision Tree, and XGBoost, as evidenced by higher accuracy, recall, sensitivity, and specificity metrics. The primary drawback of the GWO algorithm is its limited capacity to manage a large number of variables and to escape local solutions when addressing large-scale problems [56].

This paper [16] introduces the ABC-DNN model, an automated diagnostic tool for diabetes diagnosis and prediction that uses the ABC algorithm and a deep neural network. Using the Pima Indian Diabetes dataset, the model obtains an accuracy of 94.74%. A comparison with previous studies and standard DNN techniques verifies the proposed ABC-DNN model's higher performance in automated diabetes diagnosis. ABC's lack of an operator such as crossover, as seen in GA, results in an insufficient distribution of beneficial information among optimal solutions [57]. This causes the convergence performance of ABC for local minimum to be slow.

A novel soft computing model for identifying coronary artery disease in diabetic individuals is presented in this work [17]. It selects features using Rough Set (RS) theory, classifies binary data using One Versus Rest, and optimises an artificial neural network with more hidden layers using Grasshopper Optimisation Algorithm. Through single-objective optimisation, the model's accuracy increases significantly from 89.1% to 95.25%, as measured by Mean Square Error. Similar to other MH algorithms, it requires fine-tuning, and its convergence deteriorates as the dataset size increases.

Using a Radial Basis Function (RBF) neural network that has been optimised using the Genetic Algorithm (GA), this research [18] proposes a unique method for identifying heart failure. The method confirms the efficacy of its detection with 92.6% accuracy using 12 features on a heart failure dataset.

This work [19] describes an improved machine learning model for predicting heart illness and heart failure that employs meta-heuristic algorithms (CS, FPA, WOA, and HHO). Experiments on various datasets show that KNN greatly improves accuracy, demonstrating the efficiency of meta-heuristic algorithms in feature selection. The technique achieves high F-scores (99.72% for heart disease and 97.45% for heart failure), indicating significant classification accuracy improvement with these algorithms.

A metaheuristic optimisation model for automated identification of cardiac illness is presented in this work [20]. It includes an electronic low-pass filter and feature selection based on particle swarm optimisation, firefly, and cuckoo search algorithms. Using machine learning techniques including random forest, K-nearest neighbour, support vector machine, and Naive Bayes, the optimised feature vector enhances the diagnosis of cardiac disease and is evaluated on the Pascal dataset. The Firefly method exhibits inefficiencies in computational efficiency, whereas the cuckoo algorithm suffers from inadequate convergence.

The use of a Hybrid Optimisation Algorithm (HOA) to assess cardiac stroke is covered in this article [21]. The suggested method achieves a high accuracy of 97.34% by using the Crow Search Algorithm (CSA) and Whale Optimisation Algorithm (WOA) for feature extraction in a Deep Neural Network (DNN).

This research [22] provides a heart disease prediction model that combines feature extraction, record and attribute minimization using PCA, and classification with a Neural Network (NN). The model optimises weight in NN using a novel hybrid technique called Particle Swarm Optimisation merged LA update (PM-LU). PM-LU-NN outperforms other conventional algorithms in terms of accuracy, with improvements ranging from 3.85% to 12.5% over LM-NN, WOA-NN, FF-NN, PSO-NN, and LA-NN.

[23] presents a new cardiac disease prediction framework that incorporates NM-PCA for dimensionality reduction as well as an ensemble model. The optimised RNN, trained with SVM, RF, and KNN outputs, is enhanced with DUMFO, demonstrating a strong advantage over traditional models with significant accuracy gains ranging from 3.15% to 57.8%.

Ashiquzzaman A et al. [38] proposes a form of deep neural network for diabetes prognosis. In this study, the prediction system addresses the issue of overfitting is minimized by using the dropout method. Deep learning neural network is used where both fully connected layers are followed by dropout layers. This model shows the efficiency of DNN in diabetes classification task. Khanam et al. [43] also proposes a DNN for diabetes classification where it analyzes the effect of number of hidden layers on the accuracy. S Gadri et al. [44] uses a DNN with maximum of nine hidden layers with greatest depth. Theerthagiri et al. [45] uses a Multilayer Perceptron (MLP) which is a simpler version of DNN for diabetes classification.

Alam et al. [39] used multiple techniques of ANN, RF and K-means clustering for the diabetes prediction task and compared the abilities of the models proving ANN provides best accuracies. In this paper important attributes selection was done via the PCA method to improve the efficiency of the classification task.

K. Kannadasan et al. [40] utilize a DNN architecture of stacked autoencoders for feature extraction and then for classification task. The stacked autoencoder efficiently extract important features achieving an accuracy of 86.26% on the PIMA dataset.

Similarly in Garcia-Ordas et al. [41], the authors use a spare autoencoder for feature extraction and a CNN for the classification task.

Bukhari et al. [42] paper develops an improved ANN model trained using an artificial backpropagation scaled conjugate gradient neural network (ABP-SCGNN) algorithm to predict diabetes by efficiently adjusting neural network weights and biases using the scaled conjugate gradient method to minimize prediction error.

MHs are used as a helpful tool in the classification tasks. In recent years numerous new MH algorithms are released such as CWO [46], SOA [49], DOA [50], FSA [47], SBA [48], Swarm Space Hopping Algorithm (SSHA) [60], Addax Optimization Algorithm (AOA) [58], and Fossa Optimization Algorithm (FOA) [59] which include human-based, swarm-based, game-based and nature-based MH algorithms. In the current application of hyperparameter optimization of DNN for diabetes classification, these MH algorithms encounter some challenges, despite their own advantages and applications.

Unlike swarm-based methods like FSA [47], SBA [48], SSHA [60], COA incorporates both group-level coordination and individual exploration, making it more adaptable to diverse problem landscapes. Human-based algorithms, such as the CWO [46], SOA [49], and DOA [50], mimic human cognitive processes or decision-making, such as carpet weaving, sculptor making and dollmaking. These algorithms often model structured processes but may lack the stochastic adaptability inherent in COA.

The current research gap is the lack of a thorough investigation and comparison of the hybrid metaheuristics algorithms that face significant challenges including:

- A large number of MH algorithms suffer from premature convergence and result in local optima of solutions leading to suboptimal optimizations.

- Usually there is an imbalance between exploration and exploitation phase of the MH algorithm. A balance is critical for achieving both diversity (to explore the search space effectively) and accuracy (to refine the best solutions).

While earlier research has focused on various metaheuristic methods, none has examined the performance, adaptability, and generalizability of the Coati Optimisation and Cellular Automata hybrid approach to optimising DNN parameters.

The proposed COA approach has several advantages for optimization of the parameters of DNN. The first advantage of COA is that there is no control parameter in the design of this algorithm, and therefore there is no need to control the parameters in any way. COA is highly effective efficiency in dealing with a variety of optimization problems in various sciences as well as complex high-dimensional problems makes it suitable for the parameter tuning of DNN. The main advantage of COA is that it shows great ability to balance search for solutions and research in the solution search process, which allows it high-speed convergence to provide suitable values for hyperparameters in DNN optimization task which is especially a complex problem.

Cellular (CA) improves the COA by enhancing its ability to explore and exploit the search space more effectively. In CA, candidate solutions (cells) are positioned in a lattice structure, where each solution interacts with its neighbours based on predefined rules. This localized interaction allows the algorithm to refine solutions in a targeted manner, ensuring that promising areas of the search space are explored thoroughly without losing diversity. Algorithms such as CWO [46] and DOA [50] emphasize the exploration phase, drawing inspiration from artistic techniques, however may exhibit deficiencies in robust methods for exploitation or neighbourhood-based enhancements.

The suggested hybrid CCO utilizes the COA algorithm to sustain an efficient equilibrium between exploration and exploitation phases, while also capitalizing on CA's capacity to increase local neighbourhood-based improvements. CA introduces a lattice structure for organizing candidate solutions. Each cell (i.e. solution) interacts with its neighbours, encouraging diversity while converging toward the global optimum. This lattice like architecture prevents premature convergence.

Furthermore, the research gap includes a limited evaluation of the hybrid models on a variety of healthcare datasets other than PIMA Diabetes and local pathology clinics. A more thorough investigation of the algorithm's flexibility and superiority to standard Coati Optimisation approaches is required. Closing this gap would help to gain a better understanding of the algorithm's relevance and impact in both the optimisation and healthcare areas.

Table 1. Notation Nomenclature

Symbol	Meaning
Z_n	Output of n^{th} hidden layer
f, s	Activation function
A_n	Input for $(n-1)^{th}$ layer
$x_{i,j}^{P1}$	New position of i^{th} Coati
$x_{i,j}$	jth dimension of its current position
$Iguana_j$	jth dimension of the iguana's position
r	Random number in [0,1]
F_i	Objective function
lb_j^{local},ub_j^{local}	local lower bound and local upper bound of the j^{th} decision variable
F_i^{P2}	Objective function for exploitation phase
N	Number of Coati
m	Number of decision variables
$N(i)$	Neighbourhood function
°	Hadmard product
R_3	Direction coefficient
$fitness()$	Fitness value

V_i^t	Velocity of i^{th} coati at time t
P_g^t	Best global position of the i coatis
l	Number of neighbours for i^{th} coati
$t_r(\emptyset)$	Transition rule
δ_m	Distance between m^{th} and i^{th} coati
S_i	Cell state of i^{th} coati

III. DEEP NEURAL NETWORK CLASSIFIER

Deep Neural Networks (DNNs) [24] are a Deep Learning architecture that shares a basic structure with Artificial Neural Networks (ANNs). DNNs, on the other hand, distinguish themselves by incorporating numerous hidden layers, which gives them their "deep" moniker. In reality, a DNN can have up to 150 hidden layers. Each of these layers can house a large number of neurons. The input of one layer is dependent on the output of the previous layer, resulting in a sequential flow that leads to the model's ultimate prediction in the output layer.

Consider the first neuron in the hidden layer (1), which produces the output Z1. As indicated by Eq. (1), this number is the total of the products of multiple weights and inputs, as well as the bias. Z1 can span any real value from 0 to infinity, leaving the neuron unable to select whether or not to activate. Activation functions (denoted by the letter F) come into play here. They decide whether or not to activate the neuron and compute A1, which serves as the input for the next layer, and so on.

$$Z_1 = \sum_n (X_n * W_{n1}) + b$$

$$A_1 = F(Z_1) \quad (1)$$

In the proposed model, two distinct activation functions are utilised. In the hidden layers, the LeakyRelu function given by Eq. (2) is employed while the Sigmoid function given in Eq. (3) is used in the output layer, especially for binary classification tasks. To minimise output mistakes during the backpropagation process, an optimizer is required for which our proposed work uses Adamax.

$$(f(X) = \max(0.01) * X, X) \quad (2)$$

$$\left(S(x)\frac{1}{1+e^{-x}}\right) \quad (3)$$

III. COATI OPTIMIZATION

To deal with the optimisation problems, [25] proposed a new bio-inspired meta-heuristics algorithm Coati Optimization Algorithm (COA) which works in two phases:

Phase 1: Exploration Phase

This phase explores the solution space by moving coatis to different positions based on simulated interactions. This involves:

Calculation of new position (for coatis to climb the tree) using Eq. (4):

$$x_{i,j}^{P1} = x_{i,j} + r * (Iguana_j - I * x_{i,j}) \quad (4)$$

Where, where $x_{i,j}^{P1}$ is the new position computed for the ith coati, $x_{i,j}$ is the jth dimension of its current position, $Iguana_j$ is the jth dimension of the iguana's position, I is an integer randomly chosen from $\{1,2\}$, and r is a random number in the interval $[0, 1]$.

Calculating the new position (for coati that waited for iguana to fall), this is explained by Eq. (5) as,

$$x_{i,j}^{P1} = x_{i,j} + r * \left(x_{i,j} - Iguana_j^G\right) if\ objective\ function < F_i \quad (5)$$

$$x_{i,j}^{P1} = x_{i,j} + r * \left(Iguana_j^G - I * x_{i,j}\right) if\ objective\ function \geq F_i$$

When the objective function is below the threshold ($objective\ function < F_i$), the coati adjusts its position by moving away from the leader $\left(x_{i,j} - Iguana_j^G\right)$ to explore new areas. Conversely when the objective function meets or exceeds the threshold F_i, the coati moves closer to the leader $\left(Iguana_j^G - I * x_{i,j}\right)$ to exploit promising areas of the search space. This refinement focuses on intensifying the search near high-quality solutions.

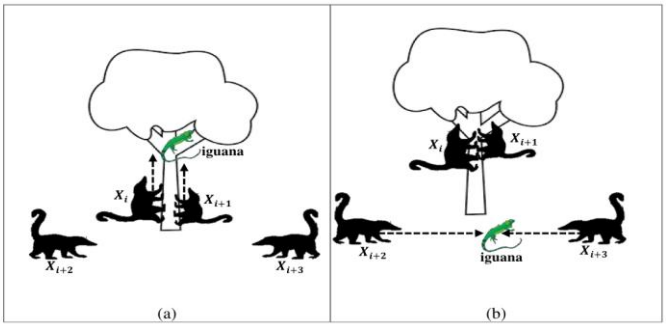

Fig. 1 A diagram showing the first phase of the COA. (a) Half of the coati population attacks the iguana on the tree. (b) The other half of the coati population hunts fallen iguanas on the ground [25].

Phase II: Exploitation Phase

Coatis move to safe positions near their current locations to avoid predators. This phase emphasizes local search and exploitation of promising areas. This phase is mathematically given in [25] as,

Generating the local and upper bounds for positions,

$$lb_j^{local} = \frac{lb_j}{t}$$

$$ub_j^{local} = \frac{ub_j}{t}$$

Where $t = 1,2,\cdots,T$, denotes the total number of iterations.

Using these, new positions are calculated as,

$$x_{i,j}^{P2} = x_{i,j} + (1 - 2r) * (lb_j^{local} + r * (ub_j^{local} - lb_j^{local}) \quad (6)$$

Then, updating the positions through objective function.

Here X_i^{P2} is the new position calculated for the i^{th} coati, based on the second phase of COA.

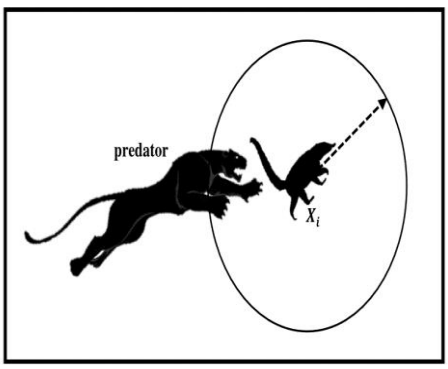

Fig. 2 Diagram showing the coati's escape from a predator during the COA second phase [25]

The method aims for a balanced approach, combining exploration and exploitation to efficiently solve optimization problems. To enhance diversity and phases, Cellular Automata (CA) is proposed to be combined with COA.

A. *Cellular Automata*

The foundational concept of Cellular Automata (CA) was introduced by [26], and its profound influence quickly spread across various engineering disciplines. The concept of a cell in CA is explained by [27] where each candidate solution in the search space is denoted as a cell. The space is represented as a virtual grid for better understanding as shown in Figure. 3. Here the smart cell is the one which represents the solution that is not sampled, i.e., has no neighbour. Hence, the position of Nth coati is defined as a state of the cell ($S_i^t = X_i$). Along with this, there are other components of CA given as, cell space, cell, cell state, transition rule, neighbourhood and discrete time steps (iterations in CO).

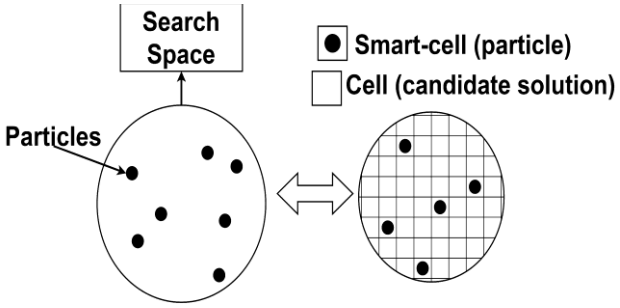

Fig. 3 Fundamental concept of Cellular Automata [26]

B. Proposed Hybridization: Coati Cellular Optimization

The hybrid Coati Optimization Algorithm with Cellular Automata is more effective at exploring and exploiting the solution space because it combines the cellular automaton components and rules inspired by COA [25]. Cells (representing solutions) update their states according to the transition rule, which takes into account data from nearby cells to further the optimization process. The incorporation of cellular automaton ideas improves COA's search capabilities and helps the algorithm locate optimal or near-optimal solutions.

The steps involved in the modelling are,

1. Initialize algorithm-specific parameters: parameters of optimization problems (range or values that each hyperparameter can take), N (number of coatis), T number of iterations, number of cell neighbours.
2. Initialize the population:

For each iteration:

2.1. Generate an initial population of individuals randomly within the search domain using equation below.

$$X_i: x_{i,j} = lb_j + r * \left(ub_j - lb_j\right), i = 1,2,3 \ldots N \; and \; j = 1,2,3 \ldots m \quad (7)$$

In this context, X_i indicates the ith coati's position in the search space, whereas $x_{i,j}$ denotes the value of the jth decision variable. The total number of coatis is denoted by N, while the number of decision variables is denoted by m. The variable 'r' represents a random real number between [0,1], and the variables lb_j and ub_j represent the lower and upper bounds of the jth decision variable, respectively.

2.2. Evaluate the Phase I and Phase II rules [25] to generate a candidate solution.

2.3. Calculate $x_{i,j}^{P1}$ based on COA Phase I or Phase II rules through Eq. (4),(5) and (6). The COA process is carried out as explained by [25].

2.4. The best candidate solution obtained through iteration process by updating the population is stored in [25].

3. CA local process activation:

3.1. The best candidate solutions are repositioned in a lattice grid, this represents a single cell.

3.2. For each cell (individual) in the lattice grid:

Initialize the neighborhood through neighborhood function given as in [27],

$$N(i)=\begin{cases} X_i^t+\dfrac{fitness(P_g^t)}{fitness(X_i^t)}R_3\circ V_i^t & fitness(X_i^t)\neq 0, fitness(P_g^t)\geq 0 \\ X_i^t+\left|\dfrac{fitness(P_g^t)}{fitness(X_i^t)}\right|R_3\circ V_i^t & fitness(X_i^t)\neq 0, fitness(P_g^t)< 0 \\ X_i^t+\left(\dfrac{e^{fitness}(P_g^t)}{e^{fitness}(X_i^t)}\right)R_3\circ V_i^t & fitness(X_i^t)\neq 0, fitness(P_g^t)\geq 0 \\ X_i^t+\left(\dfrac{e^{fitness}(P_g^t)}{e^{fitness}(X_i^t)}\right)^2 R_3\circ V_i^t & fitness(X_i^t)\neq 0, fitness(P_g^t)< 0 \end{cases} \quad (8)$$

Where R_3 is a $1\times d$ matrix formed by d uniform random numbers in $|-1,1|$ and "$\circ$" is the operation symbol of Hadmard product. We name R_3 as direction coefficient to adjust the direction and distance of the neighbourhood the jth dimension (j=1,2,….,d) of X_i^t ,N(i) generates random points within $radius_j=\left\|\frac{fitness(P_g^t)}{fitness(X_i^t)}R_{3j}V_{ij}^t\right\|, \left\|\frac{fitness(X_i^t)}{fitness(P_g^t)}R_{3j}V_{ij}^t\right\|$ or $\left\|\left(\frac{e^{fitness(P_g^t)}}{e^{fitness(X_i^t)}}\right)^2 R_{3j}V_{ij}^t\right\|$ away from X_{ij}^t . Eq. (8) governs the generation of neighbouring solutions during for i^{th} particle. It dynamically adjusts candidate positions based on fitness and energy ratios, ensuring effective exploitation of promising areas while preserving diversity. By incorporating randomization through R and directional adjustments via V, the algorithm enables localized exploration around candidate solutions. The two conditions in the equation adapt the behavior based on the quality of the parent solution, fostering a balance between exploration and exploitation. These four conditions in the Eq. (8) adapt the behavior based on the quality of the parent solution, fostering a balance between exploration and exploitation.

3.4. Neighbours generated for individual i from Eq. (8) are evaluated using the transition rule given by,

$$f(\emptyset)=\min\left(fitness(N(i)), fitness(N(i+\delta_1)), \ldots\ldots, fitness(N(i+\delta_m)), \ldots, fitness(N(i+\delta_l))\right) \quad (9)$$

$$where\ \emptyset=\begin{cases} i & if\ f(\emptyset)=fitness(N(i)), \\ i+\delta_m & if\ f(\emptyset)=fitness(N(i+\delta_m)), \end{cases}$$

$$S_i^{t+1}=S_\emptyset^t \quad (10)$$

Eq. (9) indicates that l neighbors of the ith particle, as determined by Eq. (8), are assessed, and the neighbor exhibiting the optimal fitness value is selected to substitute the i^{th} particle. The transition rule enables individuals to execute an informed jump, facilitating exploration of the search space inside a localized competitive region and augmenting the swarm's variety.Based on this, the neighbour with best fitness value is considered for replacement of i. The state of the cell is also updated. The understanding of the replacement is given as, If the fitness of the neighbour S_i^{t+1} is better than the fitness of the current position Sti, then particle i will move from its local optimal area (represented by S_i^t) to another local optimal area with better fitness. This allows the particle to explore regions of the solution space that may lead to improved solutions.

If particle j has a neighbor $S_j^{t+1}$with a better fitness than its current position S_j^t , then the transition rule causes particle j to move from its local optimal area to the global optimal area. This demonstrates how particles can move toward the best solutions in the search space. Hence, the process showcases how the combination of the neighbourhood generation, transition rule, and fitness evaluation allows particles to escape from local optima and move towards potentially better solutions, both locally and globally, in the solution space.

4. Update the current best solution if necessary.
5. Repeat until a stopping criterion is met (e.g., maximum iterations or convergence).
6. End of algorithm.

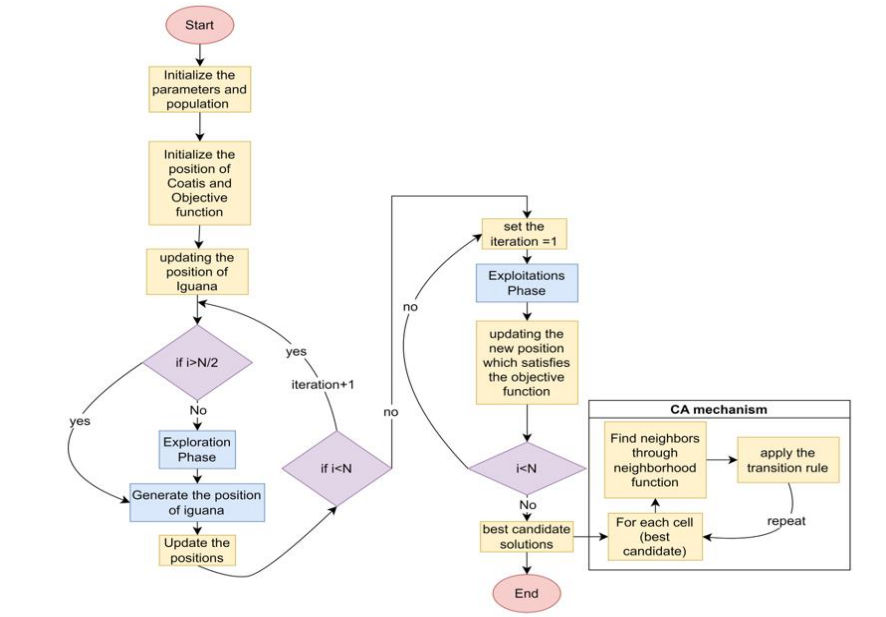

Fig. 4 Flow Chart of Coati Cellular Optimization

V. PROPOSED ALGORITHM: COATI CELLULAR OPTIMIZED DEEP NEURAL NETWORK (CCO-DNN)

The step-by-step process of our proposed Coati Cellular Optimized Deep Neural Network (CCO-DNN) classifier is as follows:

Algorithm: CCO_DNN_Optimization

The algorithm begins by defining Range of Hyperparameters, Coati Population Size (N), Maximum Iterations (T), and Neighborhood Size (l). Range of Hyperparameters include epochs, batch size, number of layers, and nodes in each layer. These hyperparameters define the DNN's structure and are crucial for its performance. N is the Number of coatis (hyperparameter optimization solutions) to evaluate in each iteration. T defines the stopping criterion for the optimization process. (l) Specifies the number of neighboring solutions (cells) to consider during the CA phase. An initial population of solutions (coatis) is generated randomly within the predefined hyperparameter ranges as given in Step 5 of the algorithm. After that the COA phase begins which operates in two primary phases to optimize the hyperparameters: exploration phase and Exploitation phase. In the exploration phase Coatis (hyperparameter solutions) explore the solution space Step 6.1, which simulate the foraging behavior of coatis. This phase ensures that diverse regions of the solution space are visited, avoiding premature convergence to local optima. In the exploitation phase coatis refine their positions near promising solutions, Step 6.2, by focusing on areas with higher fitness values. This phase concentrates on improving solutions locally to converge toward the global optimum. The fitness of each candidate solution is evaluated using the loss function. After this the CA phase of the algorithm begins and the best solutions are repositioned on a lattice grid. For each cell (solution), neighboring solutions are generated Step 7.1, where the updates are guided by the fitness values and random perturbations. The solution with the best fitness in the neighborhood is selected (Step 7.2.2), allowing the algorithm to escape local optima while refining the search around promising areas. For each candidate solution generated by the hybrid CCO optimization algorithm the DNN is trained using the hyperparameters defined by the candidate solution. The candidate solution with best fitness is retained as the optimal set of hyperparameters.

Algorithm: CCO-DNN Optimization

Input:

1. Define the range of values for hyperparameters: [epochs_range],[batch_size_range], [num_layers_range], [num_nodes_range]

2. Set the number of coatis (N)

3. Set the number of iterations (T)

4. Set the number of cell neighbors

Initialization:

5. For each coati from 1 to N:

5.1 Initialize the range of hyperparameters as Random([epochs_range],[batch_size_range], [num_layers_range], [num_nodes_range]) using Eq. (7)

COA Phase I and Phase II:

6. For each iteration from 1 to N:

6.1 Apply COA Phase I rule or Phase II rule to calculate new hyperparameter values using Eq. (4), (5), and (6) as a new range of hyperparameters

6.2 Evaluate the fitness of the candidate solution using the objective function (loss of the DNN model)

6.2.1 loss_i = TrainDNN (new_range_of_hyperparameters)

6.2.2 if loss_i < best_loss:

6.2.2.1 best_range_of_hyperparameters = new_range_of_hyperparameters

6.3 Store the best candidate solutions

CA Local Process Activation:

7. For each candidate solution (cell) in the lattice grid:

7.1 Initialize the neighborhoods of the cell using a defined neighborhood function using Eq. (8)

7.2 For each neighbor in the neighborhood:

7.2.1 Evaluate neighbors using the transition rule based on fitness values as seen in Eq. (9) and (10)

7.2.2 If the neighbor's fitness is better:

7.2.2.1 Update the candidate solution's hyperparameters

Output:

8. Optimized hyperparameters of the DNN model: best_range_of_hyperparameters

VI. Experimental Results and Analysis

A. Performance Metrics

Using stratified K-fold validation [28]**,** our proposed model (CCO-DNN) is evaluated on criteria from equation 11-16.

Accuracy (Acc) estimates a classifier's proportion of correct predictions in relation to the total number of predictions made during the testing phase.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FN+FP} * 100\% \quad (11)$$

True Positive (TP)= It indicated the number of positive cases that the model accurately categorized as positive.

True Negative (TN)= It indicates the number of negative cases accurately categorized as such by the model.

False Positive (FP)= It denotes the number of negative events that the model incorrectly forecasted as positive.

False Negative (FN)=It indicates the number of positive instances that were incorrectly predicted as negative by the model.

Sensitivity (Sens) is the fraction of True Positives classified properly by a model during the testing phase.

$$\text{Sens} = \frac{TP}{(TF+FN)} * 100\% \quad (12)$$

Specificity (Spec) is the percentage of True Negatives accurately categorised by a model during testing.

$$\text{Spec} = \frac{TN}{(TN+FP)} * 100\% \quad (13)$$

Precision (Pre) is the fraction of occurrences properly labelled as positive by a classifier compared to the total anticipated positives, showing the classifier's exactness.

$$\text{Pre} = \frac{TP}{(TP+FP)} * 100\% \quad (14)$$

Recall is computed by dividing the total number of relevant samples by the number of accurate positive results.

$$Recall = \frac{TP}{(TP+FN)} * 100\% \quad (15)$$

F1-score is the harmonic mean of precision and recall, showing their balanced combination.

$$F1 - Score = \frac{2*TP}{2*(TP+FN+FP)} * 100\% \quad (16)$$

B. Dataset Description

In this research work, we tested our proposed hybrid optimization technique on DNN using two different datasets :

a. Dataset 1: It is the PIMA Indian Diabetes dataset (PIDD) obtained via the Kaggle platform [29].

b. Dataset 2: It is made up of real data of diabetic patients obtained from a local Pathology clinic.

Dataset 1 contains 768 records, 500 of which are classed as negative, indicating non-diabetic cases, and 268 of which are classified as positive, indicating diabetes patients. This distribution accounts for 65.1% and 34.9% of the whole dataset. It has one outcome class and eight key features, which are listed in Table 2. Dataset 2 contains 80 records, with 42 cases predicted as negative (non-diabetic) and 38 cases predicted as positive (diabetes), accounting for 50.50% and 47.50% of the whole dataset. This dataset has one outcome class and is distinguished by five vital characteristics, which are listed in Table 3.

Dataset 2 is collected from a local Pathology clinic. The dataset is a real-world dataset collected from patients through collaboration with a local pathology clinic. The procedure of measuring the FBS and PPBS levels for this dataset is:

- The patient is instructed to fast (no food or drink except water) for at least 8-12 hours before the test. FBS measures blood glucose levels after the fasting period to determine baseline sugar levels. A blood sample is taken via venipuncture (drawing blood from a vein, usually in the arm). The sample is sent to the laboratory for analysis using glucose measurement devices, such as enzymatic assays.
- After the FBS test, the patient consumes a meal and wait for 2 hours after the meal before the next blood sample is taken. This period allows the blood sugar to peak following digestion and absorption of glucose from the meal. A second blood sample is taken via venipuncture. The blood sugar levels are measured in the lab, reflecting the body's ability to manage sugar after food intake.

After collecting the FBS and PPBS values, the results are recorded and interpreted based on established diagnostic thresholds. For FBS, values below 100 mg/dL are considered normal, 100–125 mg/dL indicate a prediabetic state, and levels equal to or greater than 126 mg/dL confirm diabetes. Similarly, for PPBS, values below 140 mg/dL are classified as normal, while those equal to or greater than 200 mg/dL confirm diabetes. These thresholds are used to determine the patient's diabetes status, assigning a label of 0 for non-diabetic and 1 for diabetic. It contains 80 records, with 42 cases predicted as negative (non-diabetic) and 38 cases predicted as positive (diabetes), accounting for 50.50% and 47.50% of the whole dataset. This dataset has one outcome class and is distinguished by five vital characteristics, which are listed in Table 3.

Preprocessing is crucial in data preparation for machine learning because it allows raw data to be transformed into a more acceptable format, resulting in the building of better-performing models with higher accuracy. We completed a number of critical tasks in our work, including outlier rejection [30**]**, missing value imputation [31], data balancing [32] and data normalization [33]. The quality of the data utilized for analysis is improved by all of these methods.

Table. 2 Overview of Dataset 1

Sr No	Attributes	Description
1	Pregnancies	Number of times pregnant
2	Glucose	Plasma glucose concentration 2 hours in an oral glucose tolerance test
3	Blood Pressure	Diastolic blood pressure (mm Hg)
4	Skin Thickness	Triceps skin fold thickness (mm)
5	Insulin	2-Hour serum insulin (mu U/ml)
6	BMI	Body mass index (weight in kg/(height in m)^2)
7	Diabetes Pedigree Function	Diabetes pedigree function
8	Age	Age (years)
9	Outcome	Class variable (0 or 1)

Table 3. Overview of Dataset 2

Sr No	Attributes	Description
1	Age	Age (years)
2	Gender	Gender (Male or Female)
3	Hemoglobin	Hemoglobin (g/dL)
4	FBS	Fasting Blood Sugar(mg/dL)
5	PPBS	Postprandial Blood Sugar(mg/Dl)
6	Target	Class variable (0 or 1)

C.Results and Discussion

The tests were done using a system outfitted with a Intel Core i5-1235U processor, 16 GB of RAM, and a 500 GB hard drive. The programming language was Python3, and the backend was constructed with Anaconda and implemented with Jupyter Notebook.

We used the power of hybrid Coati Cellular optimisation technology in our experiments to enhance the performance of our Deep Neural Network (DNN) technique. This hybrid technique (CCO-DNN) efficiently highlighted the most relevant hyperparameters for our model, including the number of layers, nodes in each layer, batch size, and epochs, through diligent optimization. The optimization strategy was guided by performance evaluation in the setting of stratified k-fold cross-validation.

Table 4 summarizes the results of this optimization effort, highlighting the hyperparameters that produced the best results. This strategic incorporation of hybrid Coati Cellular optimization not only improved the performance

of our DNN model but also allowed for a thorough comparison with the Coati Optimization when incorporated with DNN (CO-DNN). The goal of our research is to reduce the MSE's value stated in Eq. (17) as far as possible.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (17)$$

where y represents the actual value, $\hat{y}$ represents the predicted one, and n represents the total number of instances. Figure. 5 shows the Mean Squared Error (MSE) of CCO-DNN and the CO-DNN algorithm.

Table 4. Best Hyperparameter values

Optimization Algorithms	Best hyperparameter
CCO-DNN	Batch size=256,epochs=47,layers=4,nodes=50
CO-DNN	Batch size=256,epochs=50,layers=6,nodes=50

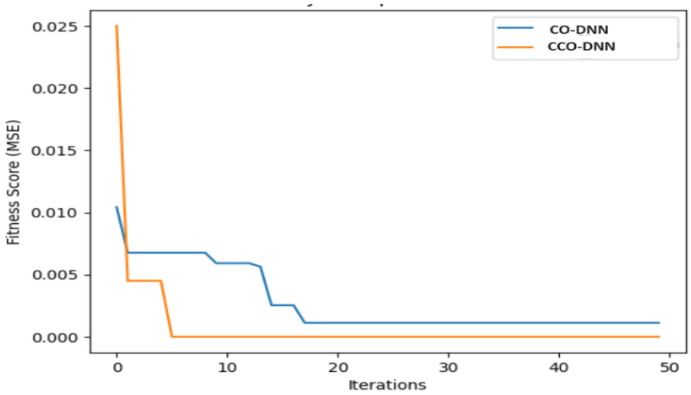

Fig. 5 CO-DNN vs CCO-DNN

Figure. 5 clearly shows that our proposed hybridization CCO-DNN has a smaller Mean Squared Error (MSE) than the CO -DNN .

Table 5. Performance of DNN using Dataset 1

Parameters	CCO-DNN	CO-DNN
Training Acc	99.24	98.23
Testing Acc	95.73	93.7
Precision	95.12	94.23
Recall	96.65	93.28
F1-Score	95.86	93.72
Spec	94.78	94.11
Sens	96.65	93.28

Table 6. Performance of DNN using Dataset 2

Parameters	CCO-DNN	CO -DNN
Training Acc	100	100
Testing Acc	99.9	98.33
Precision	100	100
Recall	100	96.66
F1-Score	100	98
Spec	100	100

According to Table 5 and Table 6, the CCO-DNN outperforms the CO-DNN across various performance metrics on the both data set.

D. Comparison of Statistical Measurement for various Classification Accuracy

The study included a thorough examination of the proposed CCO-DNN model in contrast to alternative prediction methodologies such as Random Forest [34], Naïve bayes [35], KNN [36], Adaboost [37]. The results, which are reported in Table 7 and visually displayed in Figure. 6 and Figure. 7, clearly demonstrate the better performance of the CCO-DNN over the other classifiers.

Table 7. Comparison of Performance of Proposed Model with other Machine Learning Algorithms

Parameters	NB		RF		KNN		Adaboost		Proposed Model (CCO-DNN)	
	Dataset 1	Dataset 2	Dataset 1	Dataset 2	Dataset 1	Dataset 2	Dataset 1	Dataset 2	Dataset 1	Dataset 2
Training Acc	77.51	83	100	100	85.44	100	85.42	100	99.24	100
Testing Acc	77.12	83	85.35	96.66	79.26	98.33	78.7	96.66	95.73	99.9
Precision	75.49	76.16	83.8	96	75.9	96.66	77.27	96	95.12	100
Recall	80.41	100	87.84	100	85.77	100	81.71	100	96.65	100
F1-Score	77.8	85.85	85.68	97.5	80.5	98	79.3	97.5	95.86	100
Spec	73.88	66.66	82.86	93.33	72.73	96.66	75.66	93.33	94.78	100
Sens	80.41	100	87.84	100	85.77	100	81.71	100	96.65	100

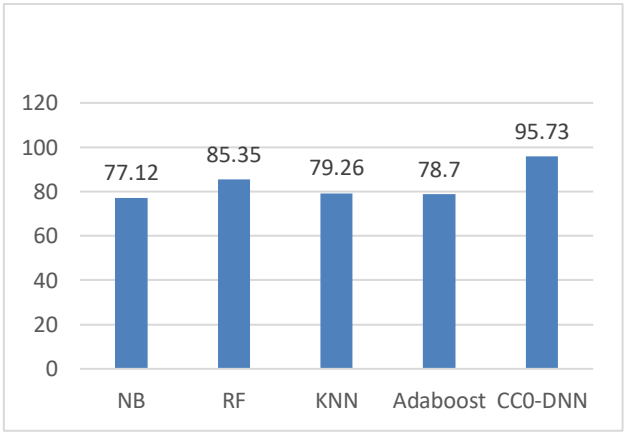

Fig. 6 Compare Proposed Model's Accuracy with other ML algorithms on Dataset 1

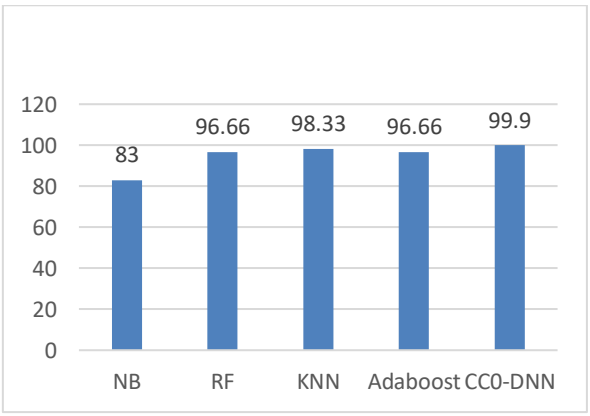

Fig. 7 Compare Proposed Model's Accuracy with other ML algorithms on Dataset 2

E. Comparative Study with other Recently Scholarly Works

The purpose of this section is to evaluate the suggested approach in view of similar studies done in the past on the same dataset as used in our work. Studies [38,45] are selected for the comparison of the proposed model. All these studies utilize Neural Network architecture for the classification of Diabetes. Another reason for selection of these datasets is that all these datasets conducted experimentation on the PIMA dataset. Papers [13,16] were also used in the comparison as these papers utilize MH optimizers for tuning along with the DL based classifiers. Results from the comparison are shown in Table 8, and as seen our proposed method achieves a higher accuracy compared to other earlier works.

Table 8. Comparison of the Proposed Model with other Existing DNN Approaches

Source	Year	Dataset	Approach	Accuracy (%)
Proposed model	2024	PIMA	CCO-DNN	95.73
Ashiquzzaman A et al. [38]	2018	PIMA	DNN	88.41
Alam et al. [39]	2019	PIMA	ANN	75.7
K. Kannadasan et al. [40]	2019	PIMA	DNN	86.26
Garcia-Ordas et al. [41]	2021	PIMA	CNN	92.31
Bukhari et al. [42]	2021	PIMA	ANN	93
Khanam et al. [43]	2021	PIMA	DNN	88.6
S Gadri et al. [44]	2021	PIMA	DNN	94.27
Theerthagiri et al. [45]	2022	PIMA	MLP	80.68
Srivastava et al. [16]	2021	PIMA	ABC-DNN	94
Zhang et al. [13]	2024	PIMA	AHDHS-stacking	93.09

Table 8 compares various diabetes classification algorithms using the PIMA dataset, with the suggested CCO-DNN model attaining the greatest accuracy of 95.73%. The data shows that classic DNN techniques, such as those of Ashiquzzaman A et al. (88.41%) and K. Kannadasan et al. (86.26%), perform poorly because to a lack of hyperparameter adjustment. Simplified models such as ANN (Alam et al., 75.7%) and MLP (Theerthagiri et al., 80.68%) produce lower accuracies due to their limited architectures, whereas more advanced approaches such as CNN (Garcia-Ordas et al., 92.31%) and ABC-DNN (Srivastava et al., 94%) perform competitively but have limitations such as insufficient hidden layers or premature convergence. Notably, models with deeper structures, such as S Gadri et al. (94.27%), improve accuracy at the expense of high computing demands. In comparison, the CCO-DNN model, optimised using hybrid tuning, strikes a compromise between performance and efficiency, obtaining the highest accuracy of 95.73%, proving its superiority in utilising a strong architecture and optimisation method.

VII. CONCLUSION

In conclusion, this work offers a unique strategy, the Coati Cellular Deep Neural Network (CCO-DNN), that uses deep learning techniques in conjunction with the Coati Optimisation Algorithm and Cellular Automata to extract significant insights from diabetes-related medical data. Extensive data preparation, validated with stratified k-fold cross-validation, considerably improves the model's predictive capabilities. The results show exceptional accuracy rates of 95.73% on the PIMA dataset and 99.9% on the real dataset. This development not only increases predictive medical modelling, but it also has the potential to improve diabetes diagnosis. This reveals that CCO outperforms

COA in terms of performance. Further by comparing our proposed approach with Coati Optimisation and other machine learning techniques, we highlight its strength and potential to revolutionise diabetes research.

DECLARATION OF COMPETING INTEREST

The authors declared no conflict of interest.

AUTHOR CONTRIBUTIONS

Sudipta Priyadarshinee contributed to conceptualization, methodology, software implementation, data curation, and manuscript writing. Madhumita Panda contributed to conceptualization, supervision, and manuscript review. All authors have read and approved the published version of the manuscript.

ACKNOWLEDGMENT

I sincerely thank my supervisor, Dr. Madhumita Panda, for her guidance and support. I also appreciate G.M. University, Sambalpur, for resources provided, and the local pathology clinic for essential data contributions.

REFERENCES

[1] V.K. Patel, V.J. Savsani, Heat transfer search (hts): a novel optimization algorithm, Inform. Sci. 324 (2015) 217–246.
[2] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948.
[3] .D.E. Goldberg, Genetic Algorithms, pearson education India, 2013.
[4] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm– a novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110 (2012) 151–166.
[5] . G.-G. Wang, S. Deb, Z. Cui, Monarch butterfly optimization, Neural Comput. Appl. 31 (2019) 1995–2014.
[6] X. Lei, X. Zhang, Y. Hao, An intelligent ellipsoid calibration method based on the grey wolf algorithm for magnetic compass, J. Bionic Eng. 18 (2021) 453–461.
[7] H. Shareef, A.A. Ibrahim, A.H. Mutlag, Lightning search algorithm, Appl. Soft Comput. 36 (2015) 315–333.
[8] 13S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. 27 (2016) 1053–1073.
[9] Zeinalnezhad, Masoomeh, and Saman Shishehchi. "An integrated data mining algorithms and meta-heuristic technique to predict the readmission risk of diabetic patients." *Healthcare Analytics* 5 (2024): 100292.
[10] Navazi, Fatemeh, Yufei Yuan, and Norm Archer. "An examination of the hybrid meta-heuristic machine learning algorithms for early diagnosis of type II diabetes using big data feature selection." *Healthcare Analytics* 4 (2023): 100227.
[11] Vijaya, J., Sanskriti Chandra, and Pallavi Baghel. "Diabetes Disease Prediction Using Various Metaheuristic Optimization Algorithms." *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*. IEEE, 2023.
[12] Li, Xiaohua, Jusheng Zhang, and Fatemeh Safara. "Improving the accuracy of diabetes diagnosis applications through a hybrid feature selection algorithm." *Neural processing letters* (2021): 1-17.
[13] Zhang, Zaiheng, et al. "A novel evolutionary ensemble prediction model using harmony search and stacking for diabetes diagnosis." *Journal of King Saud University-Computer and Information Sciences* 36.1 (2024): 101873.
[14] Naveena, Somasundaram, and Ayyasamy Bharathi. "A new design of diabetes detection and glucose level prediction using moth flame-based crow search deep learning." *Biomedical Signal Processing and Control* 77 (2022): 103748.
[15] Gadekallu, Thippa Reddy, et al. "Deep neural networks to predict diabetic retinopathy." *Journal of Ambient Intelligence and Humanized Computing* (2020): 1-14.
[16] Srivastava, Anand Kumar, Yugal Kumar, and Pradeep Kumar Singh. "Artificial bee colony and deep neural network-based diagnostic model for improving the prediction accuracy of diabetes." *International Journal of E-Health and Medical Communications (IJEHMC)* 12.2 (2021): 32-50.
[17] Kuruvilla, Ambily Merlin, and D. B. Nv. "Improved artificial neural network through metaheuristic methods and rough set theory for modern medical diagnosis." *Indian Journal of Computer Science and Engineering* 12.4 (2021): 945-954.
[18] Ay, Şevket, Ekin Ekinci, and Zeynep Garip. "A comparative analysis of meta-heuristic optimization algorithms for feature selection on ML-based classification of heart-related diseases." *The Journal of Supercomputing* (2023): 1-30.
[19] Vankadaru, Varshitha, et al. "Heart Failure Prediction Using Radial Basis with Metaheuristic Optimization." *Enabling Person-Centric Healthcare Using Ambient Assistive Technology: Personalized and Patient-Centric Healthcare Services in AAT*. Cham: Springer Nature Switzerland, 2023. 129-150.
[20] Isik, Ibrahim. "Heart Disease Prediction with Feature Selection Based on Metaheuristic Optimization Algorithms and Electronic Filter Model." *Arabian Journal for Science and Engineering* (2023): 1-14.

[21] Al-Shammari, Naif K., et al. "Cardiac stroke prediction framework using hybrid optimization algorithm under DNN." *Engineering, Technology & Applied Science Research* 11.4 (2021): 7436-7441.
[22] Cherian, Renji P., Noby Thomas, and Sunder Venkitachalam. "Weight optimized neural network for heart disease prediction using hybrid lion plus particle swarm algorithm." *Journal of Biomedical Informatics* 110 (2020): 103543.
[23] Sheeba, Paul T., Deepjyoti Roy, and Mohammad Haider Syed. "A metaheuristic-enabled training system for ensemble classification technique for heart disease prediction." *Advances in Engineering Software* 174 (2022): 103297.
[24] Gadekallu, Thippa Reddy, et al. "Deep neural networks to predict diabetic retinopathy." Journal of Ambient Intelligence and Humanized Computing (2020): 1-14.
[25] Dehghani, Mohammad, et al. "Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems." Knowledge-Based Systems 259 (2023): 110011.
[26] Schiff, Joel L. Cellular automata: a discrete view of the world. John Wiley & Sons, 2011.
[27] Shi, Yang, et al. "Cellular particle swarm optimization." Information Sciences 181.20 (2011): 4460-4493.
[28] Hasan, Md Kamrul, et al. "Diabetes prediction using ensembling of different machine learning classifiers." IEEE Access 8 (2020): 76516-76531.
[29] https://www.kaggle.com/datasets/kandij/diabetes-dataset
[30] Fitriyani, Norma Latif, et al. "Prediction Model for Type 2 Diabetes Using Stacked Ensemble Classifiers." 2020 International Conference on Decision Aid Sciences and Application (DASA). IEEE, 2020
[31] Psychogyios, Konstantinos, et al. "Missing value imputation methods for electronic health records." IEEE Access 11 (2023): 21562-21574.
[32] Priyadarshinee, Sudipta, and Madhumita Panda. "Improving prediction of chronic heart failure using smote and machine learning." *2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA)*. IEEE, 2022.
[33] Beghriche, Tawfik, et al. "An efficient prediction system for diabetes disease based on deep neural network." Complexity 2021 (2021): 1-14.
[34] Palimkar, Prajyot, Rabindra Nath Shaw, and Ankush Ghosh. "Machine learning technique to prognosis diabetes disease: Random forest classifier approach." Advanced Computing and Intelligent Technologies: Proceedings of ICACIT 2021. Springer Singapore, 2022.
[35] Priya, K. Lakshmi, et al. "A novel approach to predict diabetes by using Naive Bayes classifier." *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*. IEEE, 2020.
[36] Suriya, S., and J. Joanish Muthu. "Type 2 Diabetes Prediction using K-Nearest Neighbor Algorithm." *Journal of Trends in Computer Science and Smart Technology* 5.2 (2023): 190-205.
[37] Islam, Md Tanvir, et al. "Diabetes mellitus prediction using different ensemble machine learning approaches." 2020 11th international conference on computing, communication and networking technologies (ICCCNT). IEEE, 2020.
[38] Ashiquzzaman, Akm, et al. "Reduction of overfitting in diabetes prediction using deep learning neural network." IT Convergence and Security 2017: Volume 1. Springer Singapore, 2018.
[39] Alam, Talha Mahboob, et al. "A model for early prediction of diabetes." Informatics in Medicine Unlocked 16 (2019): 100204.
[40] Kannadasan, K., Damodar Reddy Edla, and Venkatanareshbabu Kuppili. "Type 2 diabetes data classification using stacked autoencoders in deep neural networks." Clinical Epidemiology and Global Health 7.4 (2019): 530-535.
[41] García-Ordás, María Teresa, et al. "Diabetes detection using deep learning techniques with oversampling and feature augmentation." Computer Methods and Programs in Biomedicine 202 (2021): 105968.
[42] Bukhari, Muhammad Mazhar, et al. "An improved artificial neural network model for effective diabetes prediction." Complexity 2021 (2021): 1-10.
[43] Khanam, Jobeda Jamal, and Simon Y. Foo. "A comparison of machine learning algorithms for diabetes prediction." Ict Express 7.4 (2021): 432-439.
[44] Gadri, Said. "Developing an efficient predictive model based on ml and dl approaches to detect diabetes." Informatica 45.3 (2021).
[45] Theerthagiri, Prasannavenkatesan, A. Usha Ruby, and J. Vidya. "Diagnosis and classification of the diabetes using machine learning algorithms." *SN Computer Science* 4.1 (2022): 72.
[46] Alomari, S., Kaabneh, K., AbuFalahah, I., Gochhait, S., Leonova, I., Montazeri, Z., ... & Eguchi, K. (2024). Carpet Weaver Optimization: A Novel Simple and Effective Human-Inspired Metaheuristic Algorithm. *International Journal of Intelligent Engineering & Systems*, *17*(4).
[47] Kusuma, P. D. (2024). Focus and Shake Algorithm: A New Stochastic Optimization Employing Strict and Randomized Dimension Mappings. *International Journal of Intelligent Engineering & Systems*, *17*(3).
[48] Kusuma, P. D., & Dinimaharawati, A. (2024). Swarm Bipolar Algorithm: A Metaheuristic Based on Polarization of Two Equal Size Sub Swarms. *International Journal of Intelligent Engineering & Systems*, *17*(2).

[49] Hamadneh, T., Kaabneh, K., AlSayed, O., Bektemyssova, G., Montazeri, Z., Dehghani, M., & Eguchi, K. (2024). Sculptor Optimization Algorithm: A New Human-Inspired Metaheuristic Algorithm for Solving Optimization Problems. *International Journal of Intelligent Engineering & Systems*, *17*(4).

[50] Kaabneh, K., AbuFalahah, I., Eguchi, K., Gochhait, S., Leonova, I., Montazeri, Z., & Dehghani, M. (2024). Dollmaker Optimization Algorithm: A Novel Human-Inspired Optimizer for Solving Optimization Problems. *International Journal of Intelligent Engineering & Systems*, *17*(3).

[51] Dehghani, M., Montazeri, Z., Givi, H., Guerrero, J. M., & Dhiman, G. (2020). Darts game optimizer: A new optimization technique based on darts game. *International Journal of Intelligent Engineering and Systems*, *13*(5), 286-294.

[52] Montazeri, Z., Niknam, T., Aghaei, J., Malik, O. P., Dehghani, M., & Dhiman, G. (2023). Golf optimization algorithm: A new game-based metaheuristic algorithm and its application to energy commitment problem considering resilience. *Biomimetics*, *8*(5), 386.

[53] Lambora, A., Gupta, K., & Chopra, K. (2019, February). Genetic algorithm-A literature review. In *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)* (pp. 380-384). IEEE.

[54] Gharehchopogh, F. S., & Gholizadeh, H. (2019). A comprehensive survey: Whale Optimization Algorithm and its applications. *Swarm and Evolutionary Computation*, *48*, 1-24.

[55] Abualigah, L., Diabat, A., & Geem, Z. W. (2020). A comprehensive survey of the harmony search algorithm in clustering applications. *Applied Sciences*, *10*(11), 3827.

[56] Faris, H., Aljarah, I., Al-Betar, M. A., & Mirjalili, S. (2018). Grey wolf optimizer: a review of recent variants and applications. *Neural computing and applications*, *30*, 413-435.

[57] Kumar, B., & Kumar, D. (2013). A review on Artificial Bee Colony algorithm. *International Journal of Engineering & Technology*, *2*(3), 175.

[58] Hamadneh, T., Kaabneh, K., Alssayed, O., Eguchi, K., Gochhait, S., Leonova, I., & Dehghani, M. (2024). Addax Optimization Algorithm: A Novel Nature-Inspired Optimizer for Solving Engineering Applications. *International Journal of Intelligent Engineering & Systems*, *17*(3).

[59] Hamadneh, T., Batiha, B., Werner, F., Montazeri, Z., Dehghani, M., Bektemyssova, G., & Eguchi, K. (2024). Fossa Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Engineering Applications. *International Journal of Intelligent Engineering & Systems*, *17*(5).

[60] Kusuma, P. D., & Kallista, M. (2024). Swarm Space Hopping Algorithm: A Swarm-based Stochastic Optimizer Enriched with Half Space Hopping Search. *International Journal of Intelligent Engineering & Systems*, *17*(2).

[61] Houssein, E. H., Samee, N. A., Mahmoud, N. F., & Hussain, K. (2023). Dynamic coati optimization algorithm for biomedical classification tasks. *Computers in Biology and Medicine*, *164*, 107237.