

Optimised Dendritic Cell Algorithm Using Clonal Selection Algorithm: CSA-DCA

NURA MUKHTAR
Computer Science
Umaru Musa Yaradua University
Katsina, Katsina State
NIGERIA

Abstract: - The Dendritic Cell Algorithm (DCA) is a classification algorithm inspired by the biological immune system that has shown promise for anomaly detection. However, the DCA relies on manual or domain expertise to tune signal weights which limits practical deployment and reproducibility. This paper addresses the DCA parameter optimisation challenge by applying the Clonal Selection Algorithm to optimise the weight matrix. The approach was evaluated using nine benchmark datasets achieving mean ROC-AUC of 96.48% compared to 66.56–75.33% for manually configured baselines (DCA-2 and DCA-1 individually, and best-of-the-two, respectively). CSA-DCA was also compared with alternative metaheuristic methods (GA-DCA: 92.25%; PSO-DCA: 90.83%), achieving numerically higher mean performance, though the difference was not statistically significant (Wilcoxon $p = 0.082$ vs. GA-DCA, $p = 0.213$ vs. PSO-DCA; small effect sizes, $d = 0.34$ and 0.41). The variance of CSA-DCA ($CV = 7.88\%$) was lower than both manual baselines ($CV > 35\%$), confirming more consistent performance across diverse problem domains.

Key-Words: - Dendritic Cell Algorithm; Clonal Selection Algorithm; metaheuristic optimisation; parameter tuning; anomaly detection; artificial immune systems; ROC-AUC.

Received: April 15, 2026. Revised: June 18, 2026. Accepted: July 21, 2026. Published: October 21, 2026.

1 Introduction

Artificial Immune Systems (AIS), a field that translates ideas from immunology into computational models. Artificial immune systems are computational models that draw on how the immune system is organised, how it behaves, and how it learns over time. At the conceptual level, artificial immune systems map immune processes such as negative selection, clonal expansion, affinity maturation, immune memory, and danger signalling into algorithmic forms that can be applied to computational tasks [1]. Since their introduction, AIS have been studied and applied in areas such as network intrusion detection, fault diagnosis, medical decision support, and robotics [1], [2].

The Dendritic Cell Algorithm (DCA) is an immune-inspired system proposed by Greensmith [3]. The DCA mimics the behaviour of dendritic cells. Biological dendritic cells function as context-sensitive sentinels: they collect antigens from peripheral tissues while simultaneously monitoring environmental signals that indicate tissue health or stress. Rather than reacting directly to pathogens, DCs interpret combinations of signals: namely the pathogen-associated molecular patterns (PAMP), safe signals

(SS), and danger signals (DS) to determine whether an immune response should be initiated [4]. This signal-based, context-aware behaviour provides a computationally powerful metaphor for classification tasks in which decisions must integrate multiple evidence sources of varying reliability [5]. The DCA has been applied to a variety of classification and anomaly detection problems, including network intrusion detection [6], port scan detection [7], and robotic security [8]. However, the algorithm has attracted criticism regarding its dependence on manually configured parameters. Specifically, the signal weight matrix, which derives the output cumulative values from the input signals and antigen data. The manual configuration of this parameter demands deep domain expertise and has been shown to produce highly variable and often unreliable results across different datasets [9], [10], [11].

Previous research has explored the application of metaheuristic optimisation to automate DCA weight configuration. Zhang et al. [12] developed GA-DCA using Genetic Algorithms, and Elisa et al. [13] proposed PSO-DCA using Particle Swarm Optimisation. These approaches demonstrated improvements over manual settings, but to the best of

my knowledge, no prior work has exploited the capability of the Clonal Selection Algorithm (CSA) for this purpose. The CSA is a metaheuristic whose hypermutation mechanism and affinity-proportional search strategy offer complementary properties for navigating high-dimensional, multimodal optimisation landscapes [14].

This paper proposes a novel hybrid framework integrating CSA optimisation into the DCA signal weight configuration process (CSA-DCA). The approach evolves the DCA's weight matrix through selection, cloning, and hypermutation, mimicking immune affinity maturation. The proposed approach is applied to nine different datasets for system validation and evaluation.

The rest of this paper is organised as follows. Section 2 reviews the literature on the DCA, metaheuristic optimisation, and the Clonal Selection Algorithm. Section 3 describes the proposed CSA-DCA approach. Section 4 presents the experiments and analyses the results. Section 5 draws the conclusion and identifies future research directions.

2 Literature Review

2.1 The Dendritic Cell Algorithm

The Dendritic Cell Algorithm was formally introduced by Greensmith et al. [5] as a computational model inspired by the danger theory of immune response [4]. In biological immunity, dendritic cells sample peripheral antigens while monitoring three categories of environmental signals: pathogen-associated molecular patterns (PAMPs), danger signals (DS), and safe signals (SS). These signals are processed and combined through a maturation pathway that results in either mature or semi-mature DC phenotype [5]. The computational DCA maps input data features onto these three signal categories, which are transformed through a weight matrix to produce output signals: the costimulatory molecule (CSM), the semi-mature signal, and the mature signal. These outputs accumulate within each artificial DC until a migration threshold is reached, after which the DC classifies its collected antigens based on the Mature Context Antigen Value (MCAV). An antigen is classified as anomalous if its MCAV exceeds a predefined anomaly threshold [6].

Stibor et al. [15] raised formal concerns about the DCA's sensitivity to parameter configuration, and Gu et al. [11] demonstrated analytically that DCA behaviour depends fundamentally on parameter values that cannot be determined a priori. This shows that parameter configurations producing excellent

results on one dataset frequently fail to generalise to others, necessitating repeated manual tuning.

2.1.1 Mathematical Representation of DCA DCA Signal Processing Model

For each artificial dendritic cell, input signals are represented as a vector:

$$S=[S_1, S_2, S_3] \quad (1)$$

where S_1 , S_2 and S_3 correspond to PAMP, danger, and safe signals, respectively.

Signal Transformation

The DCA processes input signals using a 3×3 weight matrix W :

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \quad (2)$$

where w_{ij} denotes the weight mapping input signal $j \in \{1, 2, 3\}$ to output signal $i \in \{1, 2, 3\}$. The output signal vector O is derived via matrix multiplication:

$$O = W \cdot S = \begin{bmatrix} O_{csm} \\ O_{sm} \\ O_{mat} \end{bmatrix} \quad (3)$$

Expanding the linear combinations yields:

$$O_{csm} = w_{11}S_1 + w_{12}S_2 + w_{13}S_3 \quad (4)$$

$$O_{sm} = w_{21}S_1 + w_{22}S_2 + w_{23}S_3 \quad (5)$$

$$O_{mat} = w_{31}S_1 + w_{32}S_2 + w_{33}S_3 \quad (6)$$

The DCA produces three output signals:

$$O = [O_{csm}, O_{sm}, O_m]$$

corresponding to costimulatory, semi-mature, and mature outputs, respectively.

Migration Condition

A dendritic cell migrates when its costimulatory signal exceeds a threshold:

$$CSM_i \geq \theta_i \quad (7)$$

where:

1. θ_i : Migration threshold of cell i

Context Assignment

Upon migration, each cell assigns a context value:

$$Context_i = \begin{cases} 1, & \text{if } M_i > SM_i \quad (\text{Anomalous}) \\ 0, & \text{otherwise} \quad (\text{Normal}) \end{cases} \quad (8)$$

Antigen Presentation and MCAV

Each antigen a_j is sampled multiple times.

$$MCAV_j = (\sum Context_i) / N_j \quad (9)$$

where:

2. $MCAV_j$: Mature Context Antigen Value of antigen j
3. N_j : Number of times antigen j is presented

Classification Rule

The final classification is given by:

$$Class(a_j) = \begin{cases} \text{Anomalous,} & \text{if } MCAV_j > \tau \\ \text{Normal,} & \text{otherwise} \end{cases} \quad (10)$$

where:

4. τ : Decision threshold (typically $\tau = 0.5$)

2.2 Metaheuristic Optimisation for DCA

Parameters

The limitations of manual DCA configuration have motivated researchers to explore automated parameter tuning. Zhang et al. [12] developed GA-DCA, applying a Genetic Algorithm to optimise the nine-weight matrix parameters while retaining default threshold values, establishing the feasibility of automated DCA weight tuning. Elisa et al. [13] proposed PSO-DCA, applying Particle Swarm Optimisation to the same weight optimisation problem. PSO-DCA provided useful insights into the DCA parameter fitness landscape, revealing that multiple distinct weight configurations can achieve similar classification performance, an evidence of a multimodal landscape with numerous local optima.

2.3 The Clonal Selection Algorithm

The Clonal Selection Algorithm, formalised as CLONALG by de Castro and Von Zuben [1], is an immune-inspired metaheuristic that models the affinity maturation process. Candidate solutions (antibodies) are ranked by fitness and cloned in proportion to their rank; clones then undergo hypermutation at rates inversely proportional to their parent's fitness. High-fitness solutions receive subtle perturbations enabling local refinement, while low-fitness solutions undergo aggressive mutation facilitating exploration of distant regions [1]. A receptor editing operator replaces the lowest-fitness antibodies with randomly generated solutions at each generation, maintaining population diversity. Despite CSA's theoretical suitability and its shared immunological inspiration with the DCA, the application of CSA to DCA parameter optimisation had not been investigated prior to this work.

3 The Proposed Method (CSA-DCA)

This section presents CSA-DCA, a novel approach that applies the Clonal Selection Algorithm to optimise the signal weight matrix of the Dendritic Cell Algorithm. The primary objective is to optimise the DCA signal weight matrix using CSA, thereby improving classification performance and eliminating the need for manual parameter tuning.

In this framework, each candidate solution (antibody) represents a DCA weight vector, while the DCA itself serves as the fitness evaluation mechanism. The optimisation process iteratively refines these weight vectors through clonal selection principles: cloning, hypermutation, and selection as illustrated in Figure 1.

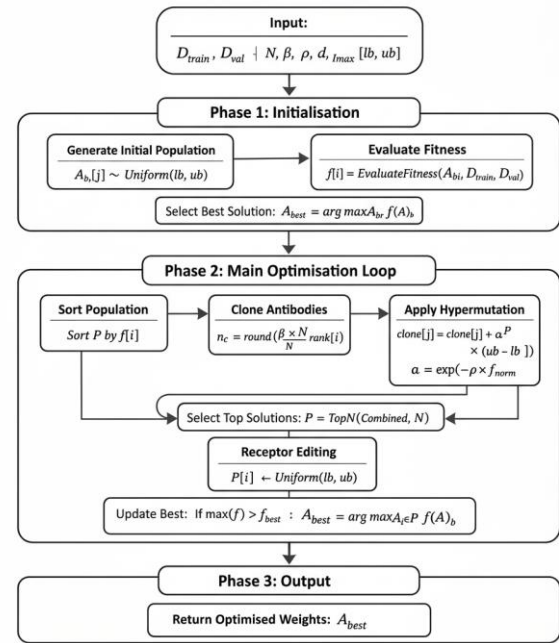


Fig. 1: CSA-DCA framework

Phase 1 Initialisation

Algorithm 1 highlights the initialisation phase of the CSA-DCA approach. Each antibody $Ab_i \in \mathbb{R}^9$ encodes one complete 3×3 DCA signal weight matrix, flattened as $[w_{11}, w_{12}, w_{13}, w_{21}, w_{22}, w_{23}, w_{31}, w_{32}, w_{33}]^T$.

Algorithm 1: CSA-DCA — Phase 1: Initialisation

Input: $D_{train}, D_{val}, N, \beta, \rho, d, I_{max}, [lb, ub] = [-10, 10], \epsilon, k$
Output: Ab_{best} — optimal 9-element DCA weight vector W^*
// Seed the population with random weight configurations
for $i \leftarrow 1$ **to** N **do**
 for $j \leftarrow 1$ **to** 9 **do**
 $Ab_i[j] \leftarrow \text{Unif}(lb, ub)$ *// each weight $\in [-10, 10]$*
 end
 $f[i] \leftarrow \text{EvalFit}(Ab_i, D_{train}, D_{val})$ *// ROC-AUC fitness evaluation*
end
// Record the initial best solution and reset counter
 $Ab_{best} \leftarrow \arg \max_i f[i]$
 $f_{best} \leftarrow \max(f)$
 $t \leftarrow 0$

The symmetric range $[-10, 10]$ admits both excitatory (positive) and inhibitory (negative) signal relationships. Fitness is evaluated immediately so that Ab_{best} is available before the optimisation loop begins.

Phase 2 Main Optimisation Loop

Algorithm 2 shows the second phase of the CSA-DCA approach, which shows the process of optimised weight matrix using CSA.

Algorithm 2: CSA-DCA — Phase 2: Main Optimisation Loop

```

while  $t < I_{\max}$  and  $|f_{\text{best}}^{(t)} - f_{\text{best}}^{(t-k)}| \geq \varepsilon$  do
  Step 1 — Fitness ranking and normalisation
  Sort  $P$  by  $f$  descending  $\Rightarrow$  assign  $\text{rank}[i] \leftarrow i$ 
   $f_{\min} \leftarrow \min(f)$ ;  $f_{\max} \leftarrow \max(f)$ 
  for  $i \leftarrow 1$  to  $N$  do
     $f^*[i] \leftarrow (f[i] - f_{\min}) / (f_{\max} - f_{\min} + \varepsilon)$  // normalise to  $[0, 1]$ 
  end
  Step 2 — Clone count allocation
  for  $i \leftarrow 1$  to  $N$  do
     $\text{nc}[i] \leftarrow \lfloor (\beta \times N) / \text{rank}[i] \rfloor$  // rank 1 receives the most clones
  end
  Step 3 — Clonal expansion and hypermutation
   $C \leftarrow \emptyset$ 
  for  $i \leftarrow 1$  to  $N$  do
    for  $k' \leftarrow 1$  to  $\text{nc}[i]$  do
       $c \leftarrow \text{Copy}(Ab_i)$ 
       $\alpha \leftarrow \exp(-\rho \times f^*[i])$  // high fitness  $\Rightarrow$  small  $\alpha$ 
      for  $j \leftarrow 1$  to  $9$  do
         $c[j] \leftarrow c[j] + \alpha \cdot N(0, 1) \cdot (ub - lb)$ 
         $c[j] \leftarrow \text{clip}(c[j], lb, ub)$ 
      end
       $f_c \leftarrow \text{EvalFit}(c, D_{\text{train}}, D_{\text{val}})$ 
       $C \leftarrow C \cup \{c\}$ 
    end
  end
  Step 4 — Elitist selection
   $P \leftarrow \text{TopN}(P \cup C, N)$  // retain  $N$  highest-fitness solutions
  Step 5 — Update global best
  if  $\max(f) > f_{\text{best}}$  then
     $Ab_{\text{best}} \leftarrow \arg \max(f)$ 
     $f_{\text{best}} \leftarrow \max(f)$ 
  end
  Step 6 — Receptor editing
  for  $i \leftarrow (N - d + 1)$  to  $N$  do
     $Ab_i \leftarrow \text{RandAb}(lb, ub)$  // replace the  $d$  weakest antibodies
     $f[i] \leftarrow \text{EvalFit}(Ab_i, D_{\text{train}}, D_{\text{val}})$ 
  end
   $t \leftarrow t + 1$ 
end

```

Fitness normalisation (Step 1). Scaling fitness to $[0, 1]$ decouples the mutation intensity from the absolute ROC-AUC scale, making the algorithm behave identically whether fitness is expressed as a probability or a percentage:

$$f_i^* = (f_i - f_{\min}) / (f_{\max} - f_{\min} + \varepsilon). \quad (11)$$

Clone count (Step 2). The rank-proportional allocation concentrates computational resources on the most promising regions:

$$\text{nc}_i = \lfloor \beta \times N / (\text{rank}(i)) \rfloor, \\ N_{\text{clones}} = \sum_{i=1}^N \text{nc}_i \approx \beta N \ln N. \quad (12)$$

Hypermutation rate (Step 3). The exponential decay creates a smooth inverse relationship between fitness and mutation intensity:

$$\alpha(f_i^*) = \exp(-\rho \cdot f_i^*), \\ \alpha \in [\exp(-\rho), 1]. \quad (13)$$

For the default $\rho = 2.5$: the best antibody receives $\alpha \approx 0.08$ (fine local tuning); the worst receives $\alpha = 1.0$ (full exploration). The per-dimension perturbation is:

$$c[j] \leftarrow c[j] + \alpha \cdot N(0, 1) \cdot (ub - lb), \\ j = 1, \dots, 9. \quad (14)$$

Elitist selection (Step 4). Combining the parent population with the clone pool and retaining the top N solutions guarantees monotonic non-decrease of the best fitness:

$$P^{(t+1)} = \text{TopN}(P^{(t)} \cup C, N), \quad (15) \\ f(Ab_{\text{best}}^{(t+1)}) \geq f(Ab_{\text{best}}^{(t)}).$$

Receptor editing (Step 6). Replacing the d lowest-fitness antibodies with uniformly random solutions prevents premature convergence by continuously injecting diversity:

$$P^{(t+1)} \leftarrow (P^{(t+1)} \setminus Ab_{\text{worst}}) \cup \\ \{Ab_{\text{new},k} \sim U(lb, ub) : k = 1, \dots, d\}. \quad (16)$$

Phase 3 Output

Algorithm 3 presents the output phase. The algorithm terminates when either (i) the generation counter reaches $I_{\max}$, or (ii) the best fitness has not improved by more than ε over the preceding k generations. The returned vector $Ab_{\text{best}} \in [-10, 10]^9$ is the 9-element weight configuration W^* that achieved the highest ROC-AUC across all evaluations. It is inserted directly into the DCA weight matrix to produce the final anomaly detector:

Algorithm 3: CSA-DCA — Phase 3: Output

```

// Termination: budget exhausted or fitness stagnated
return  $Ab_{\text{best}}$ 
// Optimal weight matrix used to configure the final DCA classifier
 $W^* \leftarrow Ab_{\text{best}}$ 
 $\hat{y} \leftarrow \text{DCA}(W^*, D_{\text{test}})$ 

```

$$W^* = Ab_{\text{best}} \Rightarrow \hat{y}_k = 1 [MCAV(a_k) > \theta_{\text{anom}}] \quad (17)$$

4 Experiment

This section presents the evaluation of CSA-DCA. The proposed CSA-DCA will be compared with five baseline methods: DCA-1 and DCA-2 (representing manually configured DCA with predefined weight matrices), GA-DCA (Genetic Algorithm optimisation) and PSO-DCA (Particle Swarm Optimisation).

This comparison will evaluate the classification accuracy and statistical significance in nine benchmark datasets, providing empirical evidence on the effectiveness of CSA for DCA parameter optimization. The experimental evaluation is as follows: Each experiment is repeated 30 times with different random seeds to account for stochastic variations in the optimisation algorithms. Also a 10-fold stratified cross-validation is employed within

each experimental run to obtain robust performance estimates. The Performance metrics are recorded for each fold and averaged to produce run-level estimates. The Final performance is reported as mean $\pm$ standard deviation across all 30 independent runs.

The experimental evaluation comprises nine benchmark datasets selected to represent diverse classification challenges commonly encountered in real world applications: Wisconsin Breast Cancer (WBC), Blood Transfusion Service Centre (BTSC), Scan, Cylinder Band, Haberman's Survival (HS), Pima Indians Diabetes (PID), Spambase (SB), Mammography (MM), and Ionosphere (IONO). These datasets span multiple domains, including medical diagnosis, telecommunications, and manufacturing quality control, and pattern recognition, with varying characteristics in terms of dimensionality, sample size, class balance, and feature structure.

4.1 CSA Configuration Parameters

Table 1. reports the CSA hyperparameter values used in all experiments. These values were selected based on established recommendations in the clonal selection literature [1] and verified through preliminary sensitivity analysis

Table 1. CSA-DCA configuration parameters

Parameter	Value	Description
Population size (N)	30	Number of antibodies
Cloning factor (β)	0.5	Controls total clone budget
Mutation decay (ρ)	2.5	Steepness of mutation–fitness curve
Receptor editing (d)	1	Antibodies replaced per generation
Max iterations (I_{max})	100	Generation budget
Convergence tolerance (ϵ)	10^{-6}	Minimum fitness improvement
Patience window (k)	10	Generations without improvement before early stop
Weight bounds [lb, ub]	[−10, 10]	Per-dimension search bounds
Cross-validation folds	10	Stratified k-fold
Independent runs	30	With different random seeds

4.2 Experimental results and analysis

This section presents performance results for the proposed CSA-DCA algorithm across nine benchmark datasets and seven evaluation metrics.

Table 2. presents the performance of CSA-DCA across all evaluation metrics. The results show consistent classification performance across diverse problem domains, with accuracy values ranging from 85.59% on the imbalanced Haberman's Survival dataset to 99.90% on the Blood Transfusion dataset. This range shows a better performance even on the most challenging problems

The CSA-DCA achieves the highest precision of 100% across five of nine datasets (WBC, Scan, Cylinder, PID, IONO), indicating low false-positive rates in these domains. The mean precision of 97.61% across all datasets indicates that when CSA-DCA classifies an instance as positive, that classification is correct in the vast majority of cases.

CSA-DCA demonstrates balanced sensitivity-specificity trade-offs on most datasets, avoiding the common pathology of sacrificing one metric for the other. The mean sensitivity (recall) of 97.06% and mean specificity of 95.17% are both high, indicating that the optimised weights produce classifiers that effectively identify both positive and negative instances.

Table 3. presents the ROC-AUC comparison between CSA-DCA and manual methods across all nine benchmark datasets. The 'Best Manual' column reports the better performance between DCA-1 and DCA-2 for each dataset, providing the strongest possible manual baseline for comparison. The final column shows the improvement achieved by CSA-DCA relative to this best manual baseline, expressed in percentage points

Table 2. Performance Results of CSA-DCA Across All Metrics (Acc. = Accuracy; Prec. = Precision; Spec. = Specificity)

Dataset	Acc.	F1-Score	Prec.	Recall	Spec.	ROC-AUC
WBC	99.86%	99.89%	100.00%	99.78%	100.00%	100.00%
BTSC	99.90%	99.90%	99.90%	99.90%	99.50%	99.70%
Scan	99.86%	99.89%	100.00%	99.78%	100.00%	100.00%
Cylinder	99.28%	99.56%	100.00%	99.12%	99.12%	99.12%
HS	85.59%	84.00%	88.20%	85.59%	64.81%	75.20%
PID	98.96%	98.48%	100.00%	97.01%	100.00%	99.78%
SB	97.67%	97.13%	94.43%	100.00%	96.16%	99.78%
MM	95.90%	95.90%	96.00%	95.90%	96.90%	96.40%
IONO	97.71%	98.18%	100.00%	96.43%	100.00%	100.00%
Mean	97.19%	96.99%	97.61%	97.06%	95.17%	96.48%
Std Dev	4.29%	4.87%	3.64%	4.36%	10.89%	7.61%

The CSA-DCA achieves the highest ROC-AUC on eight of nine datasets, with improvements exceeding 50 percentage points on three datasets: BTSC (+53.46pp), PID (+51.24pp), and Cylinder (+49.43pp). The exception is the Scan dataset, where both DCA-1 and CSA-DCA achieved a perfect 100% ROC-AUC, likely reflects DCA-1's biologically-derived default weights happen to align well with Scan's feature signal mapping. The mean improvement of CSA-DCA over the best manual configuration is 21.15 percentage points, an enhancement that changes the practical utility of DCA-based classification. Furthermore, CSA-DCA's standard deviation (7.61%) is lower than DCA-1 (26.02%) or DCA-2 (27.38%), indicating not only higher mean performance but more consistent performance across problem domains.

Table 3. ROC-AUC Performance Comparison: CSA-DCA vs Manual Methods

Dataset	DCA-1	DCA-2	Best Manual	CSA-DCA	Δ (pp)
WBC	99.79%	99.03%	99.79%	100.00%	+0.21
BTSC	46.24%	45.27%	46.24%	99.70%	+53.46
Scan	100.00%	99.68%	100.00%	100.00%	0.00
Cylinder	40.94%	49.69%	49.69%	99.12%	+49.43
HS	51.41%	44.00%	51.41%	75.20%	+23.79
PID	41.20%	48.54%	48.54%	99.78%	+51.24
SB	79.96%	87.28%	87.28%	99.78%	+12.50
MM	95.95%	26.52%	95.95%	96.40%	+0.45
IONO	85.47%	99.03%	99.03%	100.00%	+0.97
Mean	73.72%	66.56%	75.33%	96.48%	+21.15
Std Dev	26.02%	27.38%	24.05%	7.61%	–

Table 4. shows the performance comparison among optimisation methods. CSA-DCA achieves the highest mean ROC-AUC (96.48%) compared to GA-DCA (92.25%) and PSO-DCA (90.83%), representing numerical advantages of 4.23 and 5.65 percentage points, respectively. In terms of dataset-level victories, CSA-DCA wins on 5 of 9 datasets; GA-DCA wins on 1 dataset (HS); PSO-DCA wins on 2 datasets (BTSC, SB) while also tying on Scan

The result on the Cylinder Band dataset, CSA-DCA achieves 99.12% ROC-AUC while both GA-DCA and PSO-DCA achieve exactly 50.00% equivalent to random chance. This result demonstrates CSA's hypermutation mechanism escaping the majority-class trap: the 82% class imbalance creates a deceptive fitness landscape where predicting all instances as the majority class appears locally optimal. GA crossover cannot escape this trap because recombining two majority-class predictor configurations produces another majority-class predictor; PSO velocity-based search converges to the majority-class basin before discovering discriminative solutions. CSA's aggressive hypermutation of low-fitness antibodies ($\alpha \rightarrow 1$) introduces entirely novel weight values that can encode minority-class discrimination, enabling discovery of the globally optimal classifier.

Table 4. ROC-AUC Performance Comparison
Among Optimisation Methods

Dataset	CSA-DCA	GA-DCA	PSO-DCA	Best Optimiser (Margin)
WBC	100.00%	96.86%	94.44%	CSA-DCA (+3.14pp over GA)
BTSC	99.70%	95.93%	99.83%	PSO-DCA (+0.13pp over CSA)
Scan	100.00%	99.97%	99.97%	CSA-DCA (1.61pp over GA/PSO)
Cylinder	99.12%	50.00%	50.00%	CSA-DCA (+49.12pp over GA/PSO)
HS	75.20%	98.84%	92.19%	GA-DCA (+23.64pp over CSA)
PID	99.78%	97.83%	89.27%	CSA-DCA (+1.95pp over GA)
SB	99.78%	99.46%	99.96%	PSO-DCA (+0.18pp over CSA)
MM	96.40%	93.56%	95.23%	CSA-DCA (+1.17pp over PSO)
IONO	100.00%	97.80%	96.54%	CSA-DCA (+2.20pp over GA)
Mean	96.48%	92.25%	90.83%	CSA-DCA (+4.23pp over GA)
Wins	5	1 (+1 tie)	2 (+1 tie)	—

Furthermore, on the Haberman's Survival (HS) dataset, the pattern is reversed: GA-DCA achieves 98.84% ROC-AUC while CSA-DCA achieves only 75.20%, a 23.64 pp GA advantage. This dataset has extremely low dimensionality (3 features) and a moderate 2.78:1 imbalance. This suggests that the optimal weight configuration for HS requires coordinated patterns across multiple weights to combine the age, node count, and operation year features effectively. GA's crossover operation can discover and preserve such beneficial inter-weight relationships by recombining configurations that each capture part of the optimal solution. CSA's hypermutation operates independently on each weight dimension without preserving cross-weight coordination, potentially disrupting beneficial combinations once discovered. Practitioners working on similarly low-dimensional problems with complex

feature interactions should consider GA-DCA as an alternative.

4.2.1 Statistical Test Results

CSA-DCA provides more consistent performance than manual methods as presented in Table 5. CSA-DCA exhibits lower variance than both manual methods, with variance ratios of $11.69\times$ (compared to DCA-1) and $12.94\times$ (compared to DCA-2). These differences achieve statistical significance under Levene's test ($F = 12.61$, $p = 0.0027$ vs DCA-1; $F = 7.20$, $p = 0.0164$ vs DCA-2). CSA-DCA's coefficient of variation ($CV = 7.88\%$) indicates performance within approximately $\pm 8\%$ of the mean (96.48%), enabling realistic deployment planning. In contrast, DCA-1's CV of 35.30% implies performance could range from below 40% to nearly 100% with no ability to predict which outcome will occur for any given problem.

Table 5. Variance Comparison

Comparison	CSA-DCA Var	Manual Var	Ratio	Levene F	p-value
CSA-DCA vs DCA-1	57.91	677.04	$11.69\times$	12.61	0.0027* *
CSA-DCA vs DCA-2	57.91	749.67	$12.94\times$	7.20	0.0164*

Table 6. presents the Wilcoxon signed-rank test results comparing CSA-DCA against the alternative optimisation methods. The comparison with GA-DCA yields $p = 0.082$ (Wilcoxon), approaching but not reaching conventional significance at $\alpha = 0.05$. The comparison with PSO-DCA yields $p = 0.213$ (Wilcoxon), which is clearly non-significant. Effect sizes for both comparisons are small ($d = 0.34$ vs GA-DCA; $d = 0.41$ vs PSO-DCA)

These results indicate that CSA-DCA achieves higher mean performance than GA-DCA and PSO-DCA, but the differences are not statistically distinguishable from chance with nine datasets. A larger benchmark suite would be required to achieve conventional significance. This contrasts sharply with the large effect sizes ($d > 1.0$) observed in comparisons against manual methods, confirming that the fundamental performance divide lies between optimised and manual approaches rather than among different optimisation algorithms.

Table 6. CSA-DCA vs Alternative Optimizers

Comparison	W	t	p (Wilcoxon)	p (t-test)	Cohen's d
CSA-DCA vs GA-DCA	35.0	0.56	0.082†	0.589	0.34 (Small)
CSA-DCA vs PSO-DCA	30.0	0.81	0.213	0.442	0.41 (Small)

5 Conclusion

This research presents an enhancement of the Dendritic Cell Algorithm (DCA) aimed at improving anomaly detection and binary classification performance. The proposed approach applies the Clonal Selection Algorithm (CSA) to the DCA's signal weight configuration, thereby eliminating the reliance on manually defined weight matrices derived from biological intuition rather than empirical optimisation. However, this study also has certain limitations. First, while the current model maintains consistent performance across nine benchmark datasets, its effectiveness may vary when applied to large-scale or highly imbalanced real-world data where rare anomaly classes are underrepresented, as evidenced by the reduced performance observed on the Haberman's Survival dataset. Second, the current framework optimises only the nine signal weights of the DCA weight matrix, leaving the migration threshold and anomaly threshold at their default values; suboptimal threshold configurations may limit performance gains on certain problem domains.

Future research directions emerging from this work will focus on optimising signal weights, anomaly threshold, and migration threshold, and, second, on conducting a computational efficiency analysis to guide algorithm selection in time-constrained environments.

Acknowledgment:

The author would like to thank the anonymous reviewers for their constructive feedback and suggestions that significantly improved the quality of this manuscript.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the author(s) utilized artificial intelligence (AI) technologies solely for the purpose of linguistic refinement and grammatical correction. Following the application of these tools, the author(s) comprehensively reviewed, verified, and edited the resulting text. The author(s) maintain full accountability and responsibility for the accuracy, integrity, and final content of the publication.

References:

- [1] L. N. De Castro and F. J. Von Zuben, "Learning and optimization using the clonal selection principle," *IEEE transactions on evolutionary computation*, vol. 6, no. 3, pp. 239–251, 2002.
- [2] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-nonsel self discrimination in a computer," in *Proceedings of 1994 IEEE computer society symposium on research in security and privacy*, pp. 202–212, Ieee, 1994.
- [3] J. Greensmith, *The dendritic cell algorithm*. PhD thesis, University of Nottingham Nottingham, 2007.
- [4] P. Matzinger et al., "Tolerance, danger, and the extended family," *Annual review of immunology*, vol. 12, no. 1, pp. 991–1045, 1994.
- [5] J. Greensmith, U. Aickelin, and S. Cayzer, "Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection," in *International Conference on Artificial Immune Systems*, pp. 153–167, Springer, 2005.
- [6] J. Greensmith, U. Aickelin, and J. Twycross, "Articulation and clarification of the dendritic cell algorithm," in *International conference on artificial immune systems*, pp. 404–417, Springer, 2006.
- [7] J. Greensmith and U. Aickelin, "Dendritic cells for real-time anomaly detection," in *Proceedings of UK Workshop on Computational Intelligence*, (London, UK), pp. 57–66, 2007.
- [8] J. Greensmith and U. Aickelin, "Dendritic cells for real-time anomaly detection," *arXiv preprint arXiv:1001.2405*, 2010.
- [9] R. Oates, J. Greensmith, U. Aickelin, J. Garibaldi, and G. Kendall, "The application of a dendritic cell algorithm to a robotic classifier," in *International Conference on Artificial Immune Systems*, pp. 204–215, Springer, 2007.
- [10] R. Oates, G. Kendall, and J. M. Garibaldi, "Frequency analysis for dendritic cell population tuning," *Evolutionary Intelligence*, vol. 1, no. 2, pp. 145–157, 2008.
- [11] F. Gu, J. Greensmith, and U. Aickelin, "Exploration of the dendritic cell algorithm using the duration calculus," in *International Conference on Artificial Immune Systems*, pp. 54–66, Springer, 2009.
- [12] D. Zhang, Y. Zhang, and Y. Liang, "A genetic algorithm-based dendritic cell algorithm for input signal generation," *Applied*

- Intelligence, vol. 53, no. 22, pp. 27571–27588, 2023.
- [13] N. Elisa, L. Yang, F. Chao, and N. Naik, “A comparative study of genetic algorithm and particle swarm optimisation for dendritic cell algorithm,” in 2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8, IEEE, 2020.
- [14] L. N. de Castro and F. J. Von Zuben, “Learning and optimization using the clonal selection principle,” IEEE Transactions on Evolutionary Computation, vol. 6, pp. 239–251, jun 2002.
- [15] T. Stibor, R. Oates, G. Kendall, and J. M. Garibaldi, “Geometrical insights into the dendritic cell algorithm,” in Proceedings of the 11th Annual conference on Genetic and evolutionary computation, pp. 1275–1282, 2009.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Nura Mukhtar: conceptualisation, methodology, software implementation, data curation, formal analysis, original draft, review and editing.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0 https://creativecommons.org/licenses/by/4.0/deed.en_US