***Dendritic cell algorithm:***

***DC population (M) Contains:***

* Antigen Vector.
* Signal Matrix.
* Output Signals.
* Migration threshold value.

***Data Structures:***

* S = signal matrix;
* sij = a signal type i, category j in the signal matrix S;
* A = antigen vector;
* ak = antigen k in the antigen vector;
* DCm={s(m), a(m), op(m), tm}- a DC within the population;
* s(m) = signal matrix of DCm;
* a(m) = antigen vector of DCm;
* op(m) = output signal p of DCm;
* tm = migration threshold of DCm;
* wijp = transforming weight from sij op.

***Signal: Sij (read it from file contain 100 value of each)***

* **S0,0:** PAMP*-1* is the number of ICMP ‘destination unreachable’ (DU) error messages received per second.
* **S0,1:** *PAMP-2* is the number of TCP reset packets sent and received per second.
* **S1,0: (J=0, I=1)** *DS-1*, is derived from the number of network packets sent per second.
* **S1,1:** *DS-2* is derived from the ratio of TCP packets to all other packets processed by the network card of the scanning host.
* **S2,0:** *SS-1* is the rate of change of network packet sending per second.
* **S2,1:** *SS-2* is based on the observation that during SYN scans the average network packet size reduces to a size of 40 bytes, with a low standard deviation
* **S3,0:**  The inflammatory signal: is binary and based on the presence of remote root logins.

***Antigens***

* Process identification numbers (PIDs) form the antigen and are generated each time a system call is invoked (read it from this file Antigen.txt)
* Chosen Randomly from a file contain process name (max chosen 50 per DC)

***Outputs(Op):***

* **CSM(O0):** Assessed against a threshold to limit the duration of DC signal and antigen sampling, based on a migration threshold
* **Semi-Mature(O1):** Terminal state to semi-mature if greater than resultant mature signal value
* **Mature(O2):** Terminal state to mature if greater than resultant semimature signal value

***Weight Metrix: fixed number***

|  |  |  |  |
| --- | --- | --- | --- |
| **Wijp** | **J=0** | **J=1** | **J=2** |
| **P=0** | **4** | **2** | **6** |
| **P=1** | **0** | **0** | **1** |
| **P=2** | **8** | **4** | **12** |

***How it works: [Pseudocode]***

start DCA (Input Signals: *S*, Antigens: *A*)

Initialize Parameters (*I, J, M, N, O, P, Q*)

Generate *DC* population, with *M* cells

While (data evaluation)

Update (*A, S*)

For each cell *m* from *M*

Collect *Q* antigens from *A* up to *N*

For each output *p* from *O*

For each input *i* category *j* from *S*

**Op=**

If *O*0 *>* Threshold *Tm* of *DC*(*m*)

If *O*1 *>O*2

Cell context *DCm ← smDC*

Else

Cell context *DCm ← mDC*

migrate *DC*(*m*) and generate a new cell

for each antigen *a* collected from *A*

*MCAV ← mDC*(*a*)/ *smDC*(*a*)+*mDC*(*a*)

End

**The steps below show how to calculate classic DCA outputs:**

1. **Op=**

**Calculating CSM, Semi-mature and Mature outputs.**

1. DC exceeded its migration threshold **O0 >tm** And **O2>O1**

**Conclusion that “DC assigned a cell context value of “1”**, **Its collected antigen may be anomalous”**

1. DC exceeded its migration threshold **O0 >tm,** And **O2<O1**

**Conclusion that “DC assigned a cell context value of “0” Its collected antigen may be Normal”**

1. **Calculate MCAV=Mature/Antigen count**
2. Classify the result

***Example:***

This worked example consists of sample calculations for both the signal processing and antigen analysis components. For simplification, we assume a DC population size of one (M = 100). To demonstrate the calculations under different input signal conditions, 100 iterations with four sets of signals are shown.

The derived output signal values are used to demonstrate how to perform the MCAV calculation for **Four** different antigen types. In this example the following parameters apply, counting from zero.

In this example, the antigen vector (A) is updated only once at the first iteration. Each iteration will be taken independently, and calculations shown for each iterated value of “l” 100 times. 100 DCs are required in this example, one for each iteration, termed DC1 and DC2….DC100 .

Each DC is assigned an identical migration threshold value tm, to a value of **100**. The input signal values are artificially constructed so that each DC only collects one set of signals and antigen, with each DC exposed to a different set of signals.

***Parameters:***

* 100 iterations with three sets of signals.
* **K=5000**, total number of antigen used in this example with four different antigens types.
* **M=100**, DCs in the population.
* **P=2**, three output signals per DC (CSM, semi-mature, mature).
* **Q=**variable, the number of antigen sampled per iteration per DC.
* **R=50**, each DC can collect maximum 50 antigens per iteration.
* The antigen vector (A) is updated only **once** at the first iteration.
* Each iteration will be taken independently and calculations for each iteration value of **“L**
* 100 DCs are required in this example, one for each iterations DC1, DC2…DC100
* Each DC is assigned to migration threshold value **tm=100**.

**Signals:**

* PAMP -1 signal ={100 ,50,50,66,70……100} “100 value”
* PAMP -2 signal= {50 ,10,5,20,30…….100}
* Danger-1 signal={0,100,0,30,…….,100}
* Danger-2 signal={100,30,40,…..,0}
* Safe-1 signal={20,0,80,40,….100}
* Safe-2 signal={50,40,50,60…..,100}
* Inflammatory=0 or 1

S(L=0) = {S0,0=100, S0,1=50, S1,0=0, S1,1=100, S2,0=20 , S2,1=50,S3,0=1}

**Algorithm:**

* The antigen vector is updated: A= {Ag1, Ag1, Ag1, Ag2, Ag2, Ag2, Ag2, Ag3, Ag3, Ag3………Ag4………Ag4} ,**K=5000 (Ag=process name)**

The first iteration:

**Cycle L=0**

1. DC samples antigen randomly , So DC1 a(m)= {Ag1, Ag1, Ag1, Ag2, Ag2}, Chosen randomly and maximum R=50
2. DC samples input signals so DC1 S(m)= {100,50,0,100,20,50,1}
3. DC calculate output signals so DC1 Op(m):

**(CSM)O0 = [(100\*4) + (50\*2) + (0\*6) +(100\*4) +(20\*2) +(50\*6) /24] \*2 = 103.33**

The first raw from Weight matrix \* the signal matrix= Add two values /24

**(Semi)O1 = [(100\*0) + (50\*0) + (0\*1) +(100\*0) +(20\*0) +(50\*1) /2] \*2 = 50**

The second raw from Wight matrix \* the signal matrix= Add two values /2

**(Mature)O2 = [(100\*8) + (50\*4) + (0\*12) +(100\*8) +(20\*4) +(50\*12) /48] \*2 = 103.33**

The third raw from Wight matrix \* the signal matrix= Add two values/24

1. For DC1, t(m)=100

DC1 exceeded its migration threshold **O0 >tm And** **O2>O1**

**Conclusion that “DC1 assigned a cell context value of “1”** Its **collected antigen may be anomalous”**

The second iteration:

**Cycle L=1**

* **S(L=1) =** {S0,0=0, S0,1=50, S1,0=100, S1,1=100, S2,0=60 , S2,1=10, S3,0=1}

1. **DC samples antigen, So DC2 a(m) = {Ag2, Ag2, Ag3,Ag3,Ag3…Ag4}, Chosen randomly**
2. **DC samples input signals so DC2 S(m) = {0, 50, 100,100,60,10,1}**
3. **DC calculate output signals so DC2 Op(m):**

**(CSM)O0 = [(0\*4) + (50\*2) + (100\*6)+(100\*4)+(60\*2)+(10\*6) /24]\*2 = 106.67**

**(Semi)O1 = [(0\*0) + (50\*0) + (100\*1)+(100\*0)+(60\*0)+(10\*1) /2]\*2 = 110**

**(Mature)O2 = [(0\*8) + (50\*4) + (100\*12)+(100\*8)+(60\*4)+(10\*12) /48]\*2 = 106.67**

1. **For DC2, t(m)=100, DC2 exceeded its migration threshold O0 >tm , And O2<O1**

**Conclusion that “DC2 assigned a cell context value of “0” Its collected antigen may be Normal”**

**Now, we can analyses the antigens and calculate:**

MCAV=Mature/Antigen count

|  |  |  |  |
| --- | --- | --- | --- |
| **Antigen type** | **Num presentation** | **Num mature presentation** | **MCAV** |
| **Ag1** | **3** | **3** | **3/3=1** |
| **Ag2** | **4** | **2** | **2/4=0.5** |
| **Ag3** | **3** | **0** | **0/3=0** |
| **Ag4** | **----** | **----** | **-** |

To perform anomaly detection a threshold is applied to the MCAV

If we consider the anomaly threshold=0.5

**Ag1, Ag2** are classified **as anomalous,** MCAV1>0.5, MCAV2>0.5

**Ag3** classified **as Normal**, MCAV3<0.5

Ag4…..