An Immune Intelligent Approach for Security Assurance

Adriana-Cristina Enache[1], **Valentin Sgârciu**[2] and **Mihai Ioniță**[3]
University Politehnica of Bucharest, Romania[1,2]
Military Technical Academy, Bucharest, Romania[3]
adryanaenache@gmail.com[1]

Outline

1) **Introduction**
   - Intrusion Detection Overview
   - Related Works
2) **Proposed Model**
   - Our Model
   - Basic Concepts
3) **Experimental Results**
   - Model Setup
   - Results and Analysis
4) **Conclusions**

1. Introduction - Motivation

Collect data from Heterogeneous systems

Cyber threats

Security Assurance

- **Anomaly IDS may offer a viable solution**

1.1. Introduction - Overview

Intrusion Detection Systems

Their ***main role*** is to ***monitor*** the events taking place in a system, and ***determine*** if they indicate intrusions or legitimate use of the system.

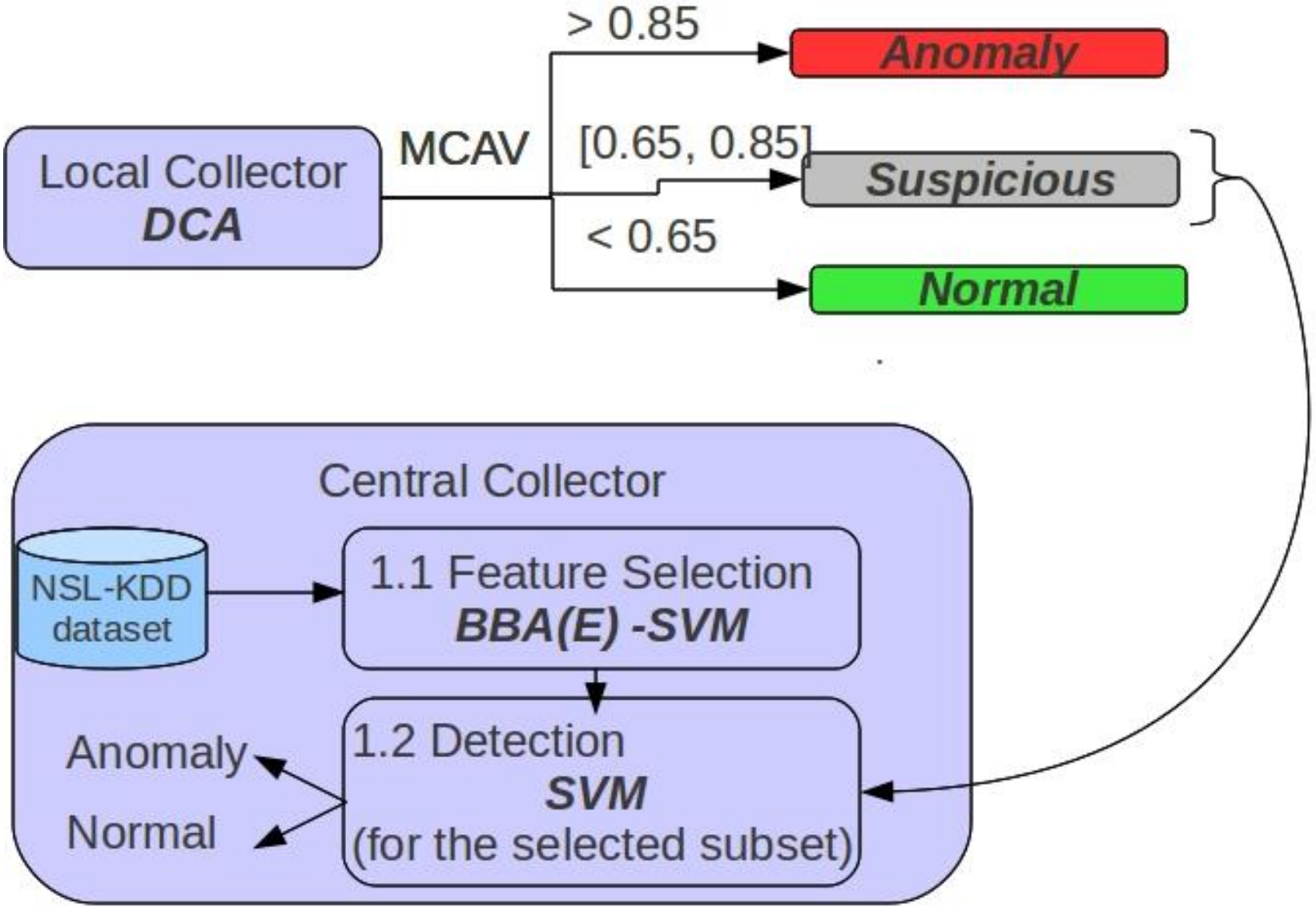
- IDS can be classified based on:
  - ✓ data analysis approach:
    - misuse (signature) detection - most widely used and it only detects known attacks.
    - anomaly detection - constructs a normal behavior profile and detects intrusions based on deviations
  - ✓ source of the analyzed information : network and host IDS.
  - ✓ reaction to an attack: active and passive

1.2. Related Works

- Recently, IDS models are based on ***computational intelligence algorithms*** :
  - ✓ **use knowledge**
  - ✓ **process large volumes of data**
  - ✓ **offer continual learning**
- **Bio-inspired algorihms** have become increasigly popular:
  - ✓ **simplicity + flexibility**

- **2008** - J. Greensmith, F. Gu and U. Aickelin – Dendritic Cell Algorithm
- **2009** - Wang et. al. - combine SVM with
  - ✓Binary PSO feature selection.
  - ✓Standard PSO for parameter optimization.
- **2010** - Wang et. al. - combine Artificial Bee Colony with SVM

2. Proposed Model



- **2 FILTERS : L.C.** (rapid + primary) **+ C.C.** (gathers suspicious data + carries out 2nd)

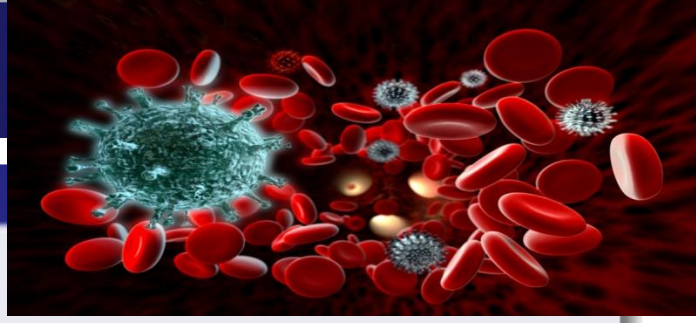
2.1. Artificial Immune System

Artificial Immune Systems

AIS is a sub-field of computational intelligence inspired by the **principals** and **functions** of the biologically immune system

- AIS properties for I.D. :
  - ✓ robustness, scalability, decentralized, adaptable to changes
- Classified:
  - ✓ Negative selection
  - ✓ Clonal selection
  - ✓ Danger theory -> **Dendritic Cell Algorithm**

2.1 Dendritic Cell Algorithm

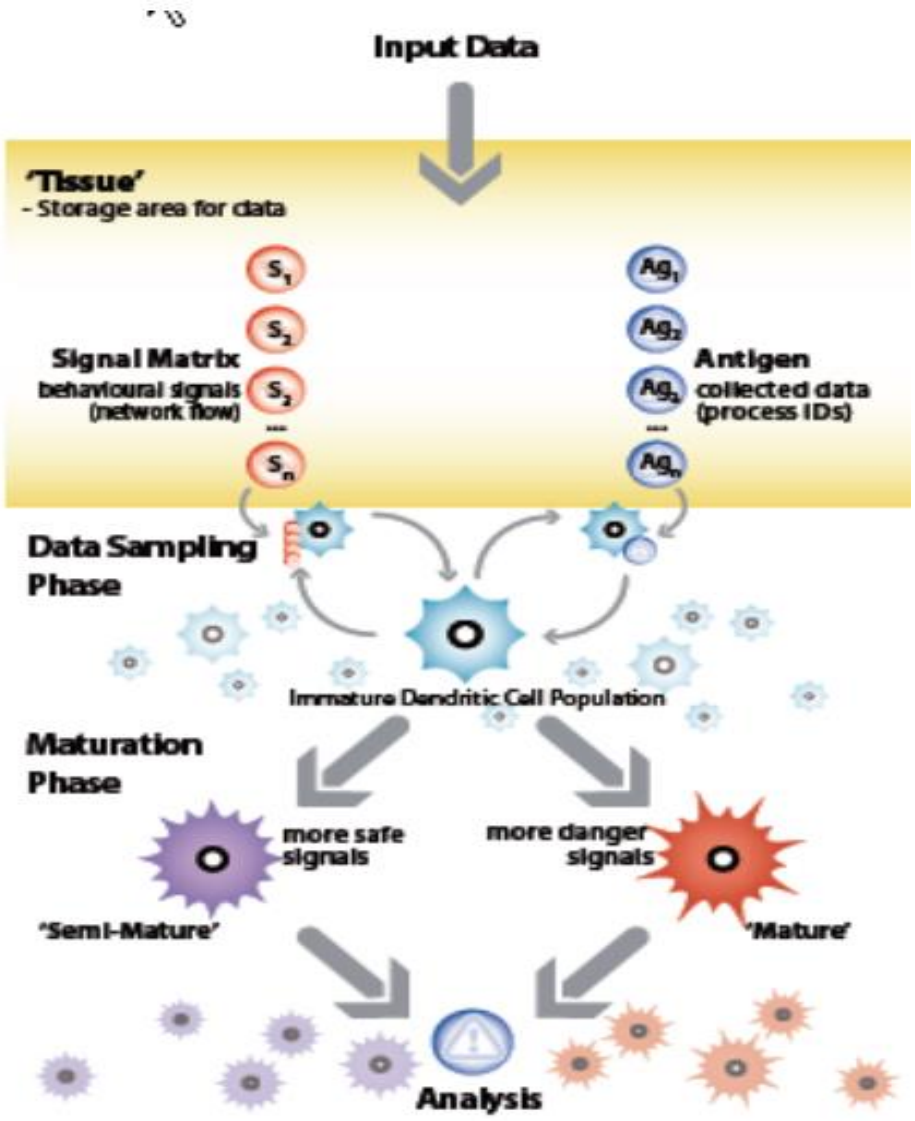


Dendritic Cell Algorithm (DCA)

mimics the behavior of dendritic cells
that respond to some specific forms of ***danger signals***

- Combining:
  - ✓ **Antigens –** the item we need to classify
  - ✓ **Signals –** evaluates the context
    - **{ PAMP | Danger | Safe }**
  - ✓ **Mature context antigen Value (MCAV)** in (0, 1)
- DCA is based on a population of DC cells:
  - ✓ **Immature** – collects antigens
  - ✓ **Semi-mature** – internally decide local signals
  - ✓ **Mature** – internally decide local signals

2.1 Dendritic Cell Algorithm



- DCA has three main stages
  - ✓ **Preprocessing**

    antigen representation + signal classification
  - ✓ **Detection**
    - sample antigen
    - **{ PAMP | Danger | Safe }**
  - ✓ **Analysis**
    - det. **MCAV** for each antigen type

2.2 Bat Algorithm

Bat Algorithm

Recently swarm intelligence algorithm [Yang2010] that was inspired from the echolocation of bats

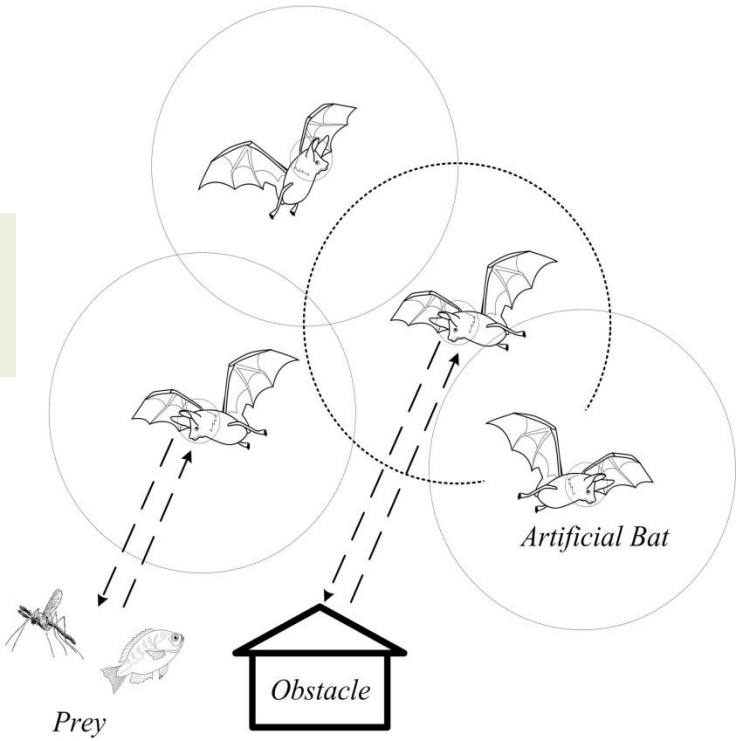
- BA performs searches:
  - ✓ **each bat** has : **location** (x_i), **velocity** (v_i), **frequency** ($freq_i$)

$$\mathrm{freq}_i = \mathrm{freq}_{min} + (\mathrm{freq}_{max} - \mathrm{freq}_{min}) \cdot \beta$$
$$\mathrm{v}_i^t = \mathrm{v}_i^{t-1} + (\mathrm{x}_i^{t-1} - \mathrm{x\_best}_j) \cdot \mathrm{freq}_i$$
$$\mathrm{x}_i^t = \mathrm{x}_i^{t-1} - \mathrm{v}_i^t$$

  - ✓ **approaches its target**: decrease **loudness +** increase **pulse rate**

$$A_i^{t+1} = \alpha\, A_i^t \qquad \mathrm{r}_i^{t+1} = \mathrm{r}_i^0 + [1 - e^{-\gamma \cdot \mathrm{t}}]$$

  - ✓ **add exploration : random walks**



- **BA** has two main components:

exploration (random walks) + exploitation (adjusting)

2.2 Bat Algorithm – Our prop. improvement

- Our ***proposed improvement*** addresses the **exploration**
- BA has a ***quick start*** + ***looses exploration*** because:

$$r_i^{t+1} = r_i^0 + [1 - e^{-\gamma \cdot t}]$$ ***=> Entering exploration :*** **Rand(0,1) > r**

- To enhance exploration, add **Euclidean distance**

$$x_{new} = x_{old} + \delta A_t^* \quad => \quad x_{new} = x_{old} + u \sqrt{\sum_{i=1}^{d} (x_{old} - x_j)^2} + \delta A_t^*$$

*x_j is a neighbor with a **better fitness function** than x_{old}*

2.2. BBA for Feature Selection

- To construct our Feat. Sel. Approach:
  - ✓ transform BA -> **BBA** by using the sigmoid function:

$$S(v_{i,j}) = \frac{1}{1 + e^{-v_{i,j}}}$$

  - ✓ the solution of *bat i* becomes:

$$x_{i,j} = \begin{cases} -1 \; if \; S(v_{i,j} > \delta) \\ 0 \; otherwise \end{cases}$$

  - ✓ assume the ***bat's position*** is d-dimensional variable

$$\boldsymbol{x_i} = (\boldsymbol{x_{i,1}}, \boldsymbol{x_{i,2}}, \dots, \boldsymbol{x_{i,d}}) = (\boldsymbol{feat_1}, \boldsymbol{feat_2}, \dots, \boldsymbol{feat_d})$$

  - ✓ to determine the ***quality of the solution*** (subset of features), the ML classier computes the **fitness function:**

$$\boldsymbol{fitness} = \mathbf{60\%}\, \boldsymbol{ADR} + \mathbf{30\%}\, \frac{\mathbf{1}}{\boldsymbol{FAR}} + \mathbf{10\%}\, \frac{\mathbf{1}}{\boldsymbol{nbFeat}}$$

3.1. Experiments – Dataset + Model setup

- To test our model we used **NSL-KDD** dataset
  - ✓ 41 features
  - ✓ 4 types of attacks: DoS, R2L, U2R and Probing
  - ✓ randomly select 9,566 records
- **Local Collector -> DCA with antigen multiplication**
- **Central Collector**
  - ✓ Feature Selection -> **BBAE-SVM**
  - ✓ Detection Stage **-> SVM**
- Implemented : our proposed BBAE, BBA and BPSO + DCA in Java: 10 individuals + 41-dimensional solution
- SVM classifier we used Weka vers 3.6.10

3.2. Experiments – Results and Analysis

-Feature Selection component of the C.C. – test results-

SI Alg	Nb. of Individ.	ADR	FAR	Nb. Feat.	Nb. Iter.
BBA(E)	**2**	**99.48**	**0.46**	**19**	**50**
BBA	2	99.38	0.52	19	**80**
BPSO	2	99.27	0.61	21	**100**
BBA(E)	**5**	**99.49**	**0.42**	**16**	**13**
BBA	5	99.48	0.44	16	**18**
BPSO	5	99.54	0.40	17	**25**
Simple SVM		**89.81**	**7.28**	**41**	

- **BBAE** *outperforms* the other SI algorithms as it ***requires fewer iterations***

3.2. Experiments – Results and Analysis

Component (Algorithm)	Data Set	ADR	FAR
Local Collector (DCA)	9,566 records with 41 feat.	61.02	4.14
Central Collector (SVM)	8,071 records with 41 feat.	98.22	1.52
Central Collector (SVM)	8,071 records with 16 feat.	99.65	0.26

- **DCA** – *suspicious data is quite high*
- **BBAE – SVM** – ***lower complexity***

4. Conclusions

- Our ***main contribution*** in this paper is the **AIDS model:**
  - ✓ **Local Collector**
  - ✓ **Central Collector**
- We *improved* BA, *created* a personal implementation and *tested* our model on the NSL-KDD dataset, which showed that our proposed approach can enhance ID
- ***Future work*** will consider:
  - ✓ improving DCA with segmentation
  - ✓ adapting our proposed model to a more “tangible” solution

!!!Thank you!!!