Clonal vs. Negative Selection in Artificial Immune Systems (AIS)

Marcellus Williams
Hampton University
100 East Queen Street
Hampton, VA 23668
1(757)316-6319
marcellus.williams@hamptonu.edu

Moayed Daniel Daneshyari
Hampton University
100 East Queen Street
Hampton, VA 23668
1(757) 728-6406
moayed.daneshyari@hamptonu.edu

ABSTRACT
In this paper, we review the bio-inspired Artificial Immune System (AIS) using two detection and selection mechanism known as negative selection and clonal selections. AIS mimic the behavior of natural immune system to find the unknown pattern that have not been seen by the system similar to what bodies would do in facing the microbial entities. We simulate the behavior of negative selection and clonal selection and compare them with each other to see the benefit of each one. Our goal is to design a system that can be utilized as an Intrusion Detection (ID) tool in networking security paradigms.

General Terms
Algorithms, Design, Security

Keywords
Artificial Immune System, Clonal Selection, Negative Selection, Computer Security

1. INTRODUCTION
Artificial immune system is the idea that the concept of the human immune system can be mimicked to be utilized in computing to defend or even heal systems [1]. The human body uses an automated process of identifying viruses and pathogens that negatively impact the body, without harming the cells that are harmless. To make this process artificial covers many aspects of computing including artificial intelligence to be able to detect and make decisions with little human interference. This artificial immune system would function ideally utilizing an artificial nervous system to detect malfunctions, and initiate the right response sequence. The human body adapts to vaccinations, which introduce a harmful virus or pathogen into the body at a weakened state to set a precedence of how the body should react if faced with the real virus or threat to the system. This process can be applied in a security context if we can automate this process, and create algorithms that introduce these afflictions to the system and automate an effective analysis of the virus so it can automatically initiate the right solution sequence.

ADMI 2016, March 31-April 3, 2016, Winston-Salem, NC, USA.

2. LITERATURE REVIEW
One such proposed algorithm to be utilized in implementing an artificial immune system is the HAIS Supervised Learning Algorithm, which is meant to imitate the adaptive response of an immune system. It contains different components of the immune system including B cells, IGs, antibodies, hyper mutation, memory cells, and affinity maturation. The B cells identify data samples of different computational afflictions; the B cells operate as the first layer and contain antibodies that capture the system pathogens for identification. Once the pathogen is identified the B cell executes the appropriate protocol to cure the machine and once this is completed the system becomes more optimized to the protocol established to defeat this particular type of ailment. [1]

This algorithm is broken down into phases that allow it to mimic the response mechanism of the human immune system. The first phase of implementation for the artificial immune system algorithm is to detect an intrusion. Similar to the body in order to be a reactive system it must detect that something is abnormal or hinders optimal performance. It is difficult because the system must have a problem set with conditions already determined that are intrusions or pathogens in medical terms to react to. [2]

$$(f \in S \cap match(f, e, I, D) = malicious) \rightarrow \delta^{+}$$

$$(f \in S \cap match(f, e, I, D) = benign) \rightarrow \delta^{-}$$

For intrusion detection you analyze the conditions in the problem and determine the severity of the affliction. If the information or input given in the problem set matches the conditions of what the immune system determines as malicious, then the automated immune system will react to the positive reading and begin to analyze the affliction to manufacture the antibodies with the appropriate signatures to effectively limit damage and deal with the attack. If the conditions are read as divine the system will dismiss the negative reading. The difficulty in this procedure is false positives and negatives as highlighted in this sample portion. There must be controls set in place to limit the possibility of false positives and negatives and confirm accurate readings. There must be a threshold in place that represents the margin of error in identifying intrusive behavior. There must be a percentage that indicates the possibility for either a positive or negative reading or the system must know, which way to lean towards based on that percentage to prevent false readings. [3]

The next portion of the algorithm consists of analyzing the properties of the ailment and manufacturing the appropriate antibodies to deal with the affliction. The antibody properties are generated pseudo randomly in a random number generator when the classification of affliction is determined. The system then tries

to match itself to the signatures of virus or intrusion based on the class of the virus. It repeats this process until the appropriate signature of antibody is found and then it replicates these antibodies to deal with the affliction. [3]

The matching phase of the artificial immune system uses mathematical formulas to calculate the difference between the intrusion or virus and the signatures stored from the vaccination phase. It uses a form of the standard deviation formula to determine the difference between the signatures of the vaccination and the intrusion or virus.

$$\rho = \frac{\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{N}(X_i - \bar{X})^2 \sum_{i=1}^{N}(Y_i - \bar{Y})^2}}$$

This is the correlation coefficient which is used in this algorithm to determine how closely related the pathogen X is to the signatures stored in the system through the vaccination phase using byte signatures to determine how far the pathogen is from the matching signature.

The Hamming distance is a common function that measures how much a bit string has been corrupted or altered during transmission. This can be applied effectively in this algorithm to determine the difference between the pathogen and the signature stored in the artificial immune system. This is used to determine the landscape affinity, which identifies the appropriate antigen and bonds with the pathogen neutralizing its effect on the natural human immune system.

The landscape affinity can be artificially implementing using a threshold or limit based on the antibody and the input of the affliction. These differences are measured using bit string differences that are either measured by the correlation coefficient or the hamming distance of the bit string. [3]

$$f_{difference} = \sum_{i=1}^{N} |X_i - Y_i|$$

Once an antigen has been matched with a pathogen the artificial immune system can heal the system. In the human body the antigen neutralizes the effects of the pathogen rendering it harmless and the body disposes of the pathogen. The concept of the artificial immune system implements these using formulas that determine the right antigen to match the pathogen impacting the system. This can be applied in a security context by improving the efficiency of intrusion detection and prevention systems. It creates a more reactive system that is able to effectively detect and defeat harmful system pathogens with little human intervention. [3]

A hybrid method was also proposed consisting of a neural system that controls the deployment of antigens and antibodies. This neural system is meant to be a more effective method of the system learning the signatures of viruses. It stores the conditions from the vaccination phase for optimal implementation of the HAIS algorithm. It is able to be more efficient spending less time analyzing the pathogen, and more time reacting and defeating the pathogen that is affecting the system.

3. METHODS

3.1 Negative Selection Algorithm

The Negative Selection Algorithm is inspired by the discriminatory behavior observed in the mammalian immune system to select and eliminate harmful pathogens from the body. Through the gene expression process the human immune system makes gene libraries that develop antibodies, which attach to pathogens and neutralize their impact on the body. The signature of the antigen is identified and the population of antibodies is consulted to find the antibody that is genetically designed to combat that antigen. [4] The negative selection algorithm has been successfully applied to detect computer viruses, and has shown several advantages over previous methods of detection for being able to constantly adapt and detect anomaly in computer systems. [3]

The Negative Selection Algorithm is designed to be a detection system for abnormal activity in computer systems.

3.1.1 Pseudo Code for Detector Generation

```
Input: Data
Output: Antibodies
Generator ←∅
While(StopCondition())
        Detector←GenerateRandDetectors()
For(Detector_i∈Antibodies)
        If(Matches(Detector_i, Data))
                Antibodies←Detector_i
                End
        End
End
```

3.1.2 Pseudo Code for Detector Application

```
Input: pathogenSamples, Antibodies
For(Inputi∈ pathogenSamples)
        Input t_class←"non-self"
        For(Detector_i∈ Antibodies)
                If(Matches(Input_i, Detector_i))
                Input t_class ←"self"
                break
                end
        end
end
```

3.1.3 Detector Generation

The first algorithm randomly creates the amount of detectors that will act as the antibodies for the artificial detection system. It is designed to mirror the genetic production of antibodies in the system by obtaining the data to create the genetic material, and store that data into each antibody. This data is the signatures of the viruses each antibody is supposed to combat. It generates a random amount of antibodies, and if the detector matches the data, then the data is stored into the antibody giving it the characteristics to be able to ward off the harmful pathogen in the artificial immune system. [2]

3.1.4 Detector Application

The second algorithm uses the output of the first algorithm to detect pathogen samples and apply the appropriate antibody to combat the pathogen that matches its signature. The pathogen sample is entered into the algorithm and the antibodies generated by the previous algorithm are supposed to detect it based on the signatures that were entered in the previous algorithm. It has to be able to distinguish between normal activity and abnormal pathogens to be effective in its detection. [4]

3.2 Clonal Selection Algorithm

The Clonal Selection Algorithm is inspired by the Clonal Selection theory of immunity. The human body has been successful at protecting itself from foreign pathogens that cause illness, this process can be automated or utilized artificially by computers in a security context to detect and prevent viruses on computer systems. [4] Artificial immune model for network Intrusion consists of three different evolutionary stages Negative Selection, clonal selection, and gene library evolution. When an animal is exposed to an antigen the bone marrow cells produce antibodies, which through mitosis divide and secrete plasma cells. These cells respond to the antigen, and store memory cells, which recognize the type of antigen after the initial exposure. These memory cells speed up the response process by reacting to this pathogen faster and skipping the detection phase of the response. This can be applied to an artificial immune system by creating artificial cells that detect the pathogen, and clone artificial antibodies to overwhelm the pathogen and execute the artificial response to eliminating the virus. This has many security implications that include creating an adapted system that requires little human intervention. The system is able to conduct self-detection, and generate an automated response to return to a normal state. [5]

The Clonal Selection algorithm is designed to detect the negative conditions determined by the negative selection algorithm and begins the mass production of antibodies. It clones the existing antibodies to combat the antigens.

Clonal Selection Algorithm Pseudo Code

Input: $Population_{size}$, $Selection_{size}$, $Problem_{size}$, $RandomCells_{num}$, $Clone_{rate}$, $Mutation_{rate}$
Output: Population
Population←CreateRandomCells($Population_{size}$, $Problem_{size}$)
While(StopCondition())
    For($\rho_i \in$ Population)
        Affinity(ρ_i)
    End
$Population_{select}$←Select(Population, $Selection_{size}$)
$Population_{clones}$←∅
For($\rho_i \in Population_{select}$)
    $Population_{clones}$←clone(ρ_i, $Clone_{rate}$)
End
For($\rho_i \in population_{clones}$)
    Hypermutate(ρ_i, $Mutation_{rate}$)
    Affinity(ρ_i)
End
Population←Select(Population, $Population_{clones}$, $Population_{size}$)
$Population_{rand}$←CreateRandomCells($RandomCells_{num}$)
Replace(Population, $Population_{rand}$)
End
Return(Population)

This algorithm starts off with a population size, selection size, problem size, random cells, clone rate, and mutation rate. The algorithm executes based on the clone rate, which clones normal cells, and the mutation rate clones cells with mutations that may make it better equipped to fight off the infection. Once it is completed it returns the new population size of the antibodies. These antibodies will be used in the artificial immune system to recognize the traits of the antigens and orchestrate a response for the computer system [2].

4. SIMULATION RESULTS AND COMPARISON

In this paper, we have compared the clonal selection algorithm and negative selection algorithm in Artificial Immune System (AIS). We have simulated each algorithm separately and tested them in detecting specific cases.

The negative selection algorithm is designed to detect changes. 300 detectors are utilized to evaluate 150 self-patterns, and the conditions are returned as predicted and expected. The predicted value is based on the conditions of the data stream falling into the self or nonself categories. The expected value is what the generator returns after assessing the conditions of the system. The self-category represents benign data patterns or the computer is in a healthy normal or self-state. The N or nonself condition is a harmful data pattern developed by the system that does not fall in the range of 0.5 to 1. The negative selection algorithm randomly generates 300 detectors, and screens their ability to accurately detect the self or benign data patterns, against the non-self or abnormal data patterns. The input of the data is the problem size or bounds for the search space, and self-space. The max number of detectors allowed is 300. The max number of self or harmless data patterns is 150, and the negative selection algorithm generates these patterns. The data streams whose Euclidean distance is in between 0.5-1.0 are classified as self-data patterns, and any other distance within the 0.0 to 1.0 search space is non-self or malicious data patterns. The output is the trial number of the algorithm, the predicted results based on the algorithms classification of the randomly generated data stream within the search space, and the expected value returned by the generator created within the value. These generators are created with the self-classification, and are designed to detect which data streams are self and if the patterns match the self-condition it is supposed to return the self-category for the expected value. If there is a discrepancy between the predicted value and the expected value, that generator is ineffective. That represents a false positive within the algorithm, which makes it inaccurate for assessing, analyzing, and reacting to abnormal conditions to be effectively used for security purposes, more specifically an intrusion detection system.

We know that it is important for the human immune system to be able to replicate the most effective and adaptive cells capable of recognizing different pathogens within the body. The clonal selection algorithm automates this process. The input of the algorithm is the existing population of antibodies, which are stored in a vector. This population is immature, which means they haven't attached to an antigen within the body. Within this population the best are selected based on their ability to match against antigen patterns. The problem size is the bounds of the array that the data is stored, which are -5, and 5. The affinity rate or ability to bind to the antigens is calculated, and the selected population of antibodies is cloned to a .1 rate. The resulting population will then compete with the existing remaining antibodies in the population set for existence in the next generation. Randomly generated antibodies, to maximize the effectiveness of the next generation, replace low affinity antibodies or inefficient antibodies. This process is repeated for 50 generations, but around the 23rd generation the algorithm has

reached maximum efficiency and the affinity rate is at the highest possible meaning the shortest amount of time for the population to recognize and initiate a response to malicious data. The output is the generation, and total cost or time and resources it takes for that generation to recognize and react to malicious data, or harmful conditions within the computer system. At the 23rd generation the maximum amount of B or memory cells have been manufactured from the existing population to make the response as quick, and cost-effective as possible.

Trial	**Predicted**	**Expected**	Trial	**Predicted**	**Expected**	Trial	**Predicted**	**Expected**
1	S	S	18	N	N	35	N	N
2	N	N	19	N	N	36	S	S
3	**S**	**N**	20	S	S	37	N	N
4	N	N	21	N	N	38	N	N
5	N	N	22	N	N	39	N	N
6	S	S	23	S	S	40	N	N
7	N	N	24	S	S	41	N	N
8	N	N	25	N	N	42	N	N
9	N	N	26	N	N	43	N	N
10	N	N	27	**N**	**S**	44	N	N
11	N	N	28	N	N	45	N	N
12	N	N	29	N	N	46	N	N
13	N	N	30	N	N	47	N	N
14	**S**	**N**	31	**S**	**N**	48	**S**	**N**
15	N	N	32	N	N	49	S	S
16	N	N	33	**S**	**N**	50	N	N
17	N	N	34	S	S			

Table 1. Negative Selection for different trials; N is for normal condition, and S is for self-condition. The table shows that 44 out of 50 cases will result correct prediction.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have reviewed and compared two selection mechanisms used in natural immune system known as negative selection and clonal selection. The clonal selection algorithm demonstrates results of cloning and mutation of the antibodies while they compete to exist in the next generation of antibodies. Low affinity antibodies are supposed to be replaced with the better ones. Inspired by the natural immune system, this system is simulated using 100 generations, until a population size, and number of cells that have been mutated, and a ratio of the number of the mutated over the non-mutated cells.

The negative selection algorithm which works based upon the selection of negative genes is designed to detect changes. We showed that negative selection algorithm predicted about 88% in a trial of 50 cases. Our purpose is to use these algorithms to detect the intruders to the network.

In comparison of the algorithm, clonal selection has shown more robustness and quicker settlement into the final state compared to the negative selection. In our future work, we would input the patterns of actual intruders to different types of networks into the AIS for each different defense/detection algorithm to see the robustness of each method in real world application of network security.

6. ACKNOWLEDGMENTS

This work in part was supported by National Science Foundation (NSF) CyberCorps: Scholarship for Service grant that will provide support for students to study in the Master of Science in Information Assurance program.

Gen.	***f***	**Gen.**	***f***
1	0.012017373	26	1.16E-08
2	0.009868886	27	1.16E-08
3	0.000957021	28	1.16E-08
4	9.94E-05	29	1.16E-08
5	4.35E-05	30	1.16E-08
6	3.70E-05	31	1.16E-08
7	1.05E-05	32	1.16E-08
8	7.60E-06	33	1.16E-08
9	1.18E-06	34	1.16E-08
10	1.04E-06	35	1.16E-08
11	1.04E-06	36	1.16E-08
12	5.70E-07	37	1.16E-08
13	5.70E-07	38	1.16E-08
14	1.98E-07	39	1.16E-08
15	1.98E-07	40	1.16E-08
16	5.82E-08	41	1.16E-08
17	5.82E-08	42	1.16E-08
18	5.82E-08	43	1.16E-08
19	5.82E-08	44	1.16E-08
20	5.82E-08	45	1.16E-08
21	5.82E-08	46	1.16E-08
22	5.82E-08	47	1.16E-08
23	1.16E-08	48	1.16E-08
24	1.16E-08	49	1.16E-08
25	1.16E-08	50	1.16E-08

Table 2. Clonal Selection for 50 generations

7. REFERENCES

[1] W. Ahmad and A. Narayanan, “Principles and Methods of Artificial Immune System Vaccination of Learning Systems,” Lecture Notes in Computer Science Artificial Immune Systems, pp. 268–28

[2] J. Brownlee, Clever algorithms. Melbourne, Australia: Jason Brownlee, 2011.

[3] P. Harmer, P. Williams, G. Gunsch, and G. Lamont, “An artificial immune system architecture for computer security applications,” IEEE Transactions on Evolutionary Computation IEEE Trans. Evol. Computat., pp. 252–280, 2002.

[4] J. Kim and P. Bentley, An Evaluation of Negative Selection in an Artificial Immune System for Network Intrusion Detection, 1st ed. London: University College London, 2001, pp. 1-8.

[5] F. Von Zuben and L. De Castro, The Clonal Selection Algorithm with Engineering Applications, 1st ed. Capinas Brazil: State University of Campinas, pp. 1-7.