Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

A ESCOLHER

João Paulo C. de Araujo

Dissertação apresentada como requisito parcial para qualificação do Mestrado em Informática

Orientador
Prof.a Dr.a Genaína Nunes Rodrigues

Brasília
2022

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

A ESCOLHER

João Paulo C. de Araujo

Dissertação apresentada como requisito parcial para
qualificação do Mestrado em Informática

Prof.a Dr.a Genaína Nunes Rodrigues (Orientador)
CIC/UnB

Prof. Dr. Donald Knuth
Stanford University

Dr. Leslie Lamport
Microsoft Research

Prof.a Dr.a Ricardo Pezzuol Jacobi
Coordenadora do Programa de Pós-graduação em Informática

Brasília, 05 de Novembro de 2022

Resumo

Donec pellentesque, libero eu fermentum posuere, odio sapien blandit lacus, non tincidunt nibh mi id ante. Donec ac ligula sed eros condimentum dapibus molestie ultricies dui. Vestibulum nec luctus urna, ac finibus diam. Vivamus sapien lacus, pulvinar vel turpis sit amet, faucibus efficitur libero. Aliquam non nibh feugiat, luctus eros ac, pulvinar enim. Vestibulum a sem ac lacus tincidunt luctus. Ut dapibus, metus et consectetur venenatis, felis arcu ullamcorper neque, ultrices euismod nunc nibh in lectus. Suspendisse sem eros, lobortis ac tellus molestie, vestibulum rutrum mauris. Nunc congue pharetra erat. Mauris mi augue, euismod ut facilisis quis, rhoncus in velit. Vestibulum finibus tempus pulvinar. Mauris sodales sed urna vitae mattis.

Palavras-chave: LaTeX, metodologia científica, trabalho de conclusão de curso

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent placerat sollicitudin neque nec vehicula. Nam nec cursus ipsum. Pellentesque finibus urna vel metus egestas, eget sagittis nibh aliquet. Nulla auctor lacus a eleifend blandit. Phasellus finibus risus ac sem venenatis hendrerit. Donec ut mattis dolor, sed aliquet urna. Integer nunc orci, feugiat at mi nec, facilisis luctus neque. Suspendisse potenti. Pellentesque diam ante, vestibulum nec risus at, hendrerit fringilla urna. Aliquam suscipit tempus elit, in tincidunt nulla tincidunt sed.

In aliquet commodo iaculis. Aenean sollicitudin odio et turpis auctor suscipit. Maecenas finibus turpis felis, a ornare lorem posuere sit amet. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed tempor dignissim risus nec elementum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Ut euismod nisl at neque elementum, at imperdiet diam faucibus. Praesent a est sem. Fusce suscipit tellus id pulvinar ullamcorper. Nullam a felis et nibh dignissim tempor at quis tortor. Vestibulum vestibulum molestie pellentesque. Phasellus ac laoreet magna. Etiam est metus, consequat at pretium sit amet, auctor sit amet nulla. Nullam ligula odio, porttitor et eros id, finibus vestibulum orci.

Keywords: LaTeX, scientific method, thesis

Sumário

Lista de Figuras

Capítulo 1

Background

1.1 Artificial Immune Systems (AIS)

The primary goal of the Biological Immune System (BIS) is to protect the body from potentially harmful material [4]. It is composed by a multilevel defense mechanism that is capable of distinguishing between molecules from the body itself and foreign ones, selecting a specific response to a threat, and enacting an inflammatory reaction in order to maintain the health and safety of the body. All of these activities are capable of evolving through time since they rely on aspects of learning and memory, which are present in the BIS.

In 1986, Farmer et al. [5] sought inspiration in the properties and theories of this biological system to propose a computational model, named Artificial Immune System (AIS). It was built in such a way that the immunological concepts and processes are distilled into algorithms that can be simulated by a computer in order to solve all sorts of problems from the real world. The idea behind it is that, since this system is able to protect us, it might be computationally useful [6]. Nevertheless, by that time, the resources available were not powerful enough to allow its usage in real systems [7].

Recently, the advances in technology enabled complex tasks to be performed in a fast and efficient way. This provided the researchers the tools needed to implement and further improve the application of the immunological concepts into a wide range of problems. For example, AIS techniques have been used in problems like anomaly detection, optimization, classification and clustering [4]. The Negative Selection, along with the Clonal Selection, Immune Network Theory and the Danger Theory are amongst the most researched approaches in the literature [2].

This section is structured as follows: initially, the main concepts of the Biological Immune System will be laid out. Then, the Negative Selection algorithm will be explained in depth, for it will be exercised throughout this work.

1.1.1 Overview of the Biological Immune System

As mentioned previously, the Artificial Immune System is a model inspired by its biological counterpart in order to solve computational problems. Therefore, there is a need to introduce the main concepts and processes found in the Biological Immune System so that the analogies and metaphors implemented by the AIS might be better understood. This subsection focuses on providing the reader with the foundational knowledge on the BIS.

Main Concepts

Immunity can be defined as the ability to respond to unknown substances [8]. In this sense, the Biological Immune System (BIS) comprises a set of structures and mechanisms which are capable of distinguishing the body's cells from foreign substances and responding adequately. It includes specific organs, cells and molecules. It is a complex, fault-tolerant and distributed multilevel defense mechanism composed of two main layers, namely the innate immunity and the adaptive immunity [1].

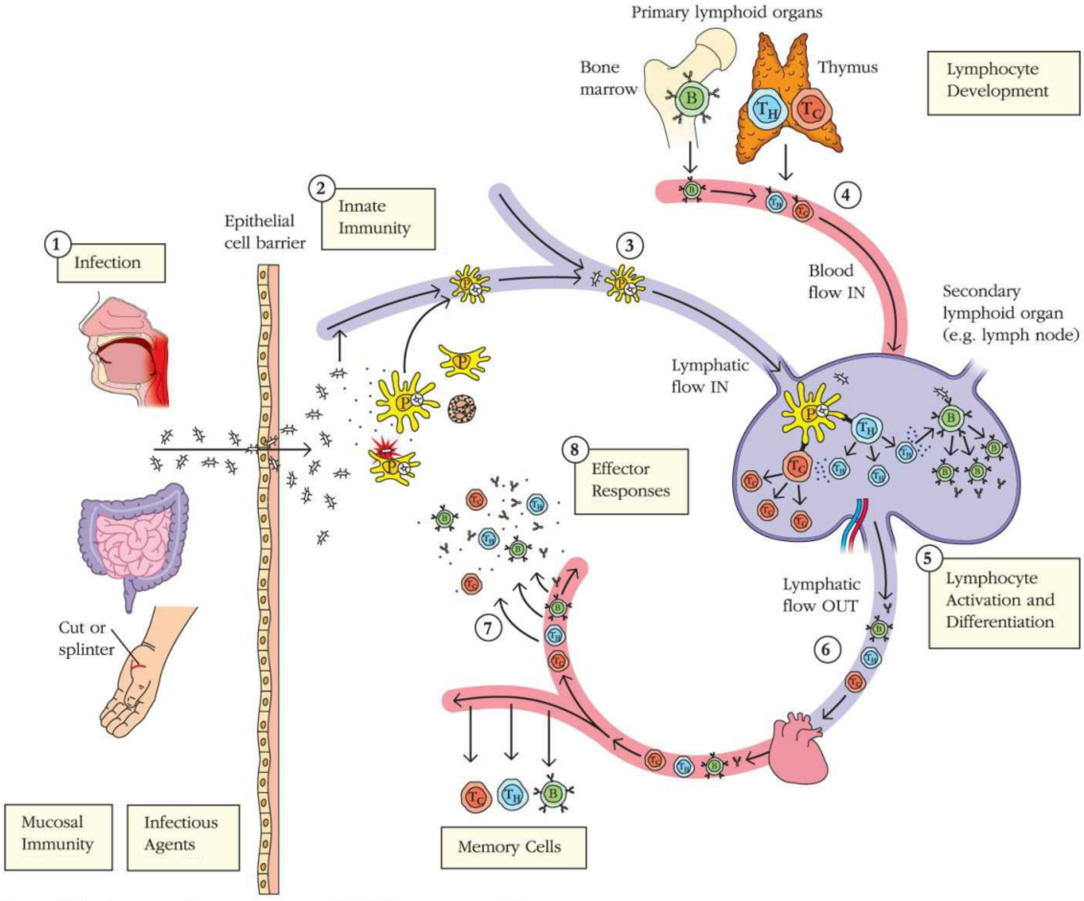
The innate immunity is the body's first line of defense. Inherited from the host's progenitors, it is responsible for a quick or immediate response against infections. It is achieved through physical and chemical barriers or through cellular responses. The barriers are considered nonspecific mechanisms that work as shields against pathogens by blocking them from entering the body. They comprise the skin, mucous membranes on the body's openings and the secretions of both. The low pH of the skin, for instance, inhibits the proliferation and growth of bacteria, while the antimicrobial substances present in saliva and tears keep the antigens from invading through the membranes [8]. The cellular response, on the other hand, focus on perceiving the pathogens that were able to surpass the barriers and activating a variety of cellular responses, which include: the ingestion of the substance (phagocytosis), the induction of an inflammatory response and the triggering of the adaptive immunity for a tailor made response.

The adaptive immunity is an immunological mechanism that is capable of "specifically recognize and selectively eliminate foreign microorganisms and molecules"[8]. In contrast with the innate immunity, it has a high level of specificity when dealing with the antigens, meaning that the response is customized and based on the particularities of the foreign substance. The downside is that the response can take days to be performed. Nevertheless, the information from previous infections is persisted in order to achieve a faster response when a similar antigen is detected. In the literature, the adaptive immunity responses is divided into two distinct, but overlapping, categories: the humoral immunity and the cellular immunity.

The first kind of response, called humoral immunity, relies on the interaction between the antigen and B lymphocytes, a specific type of white blood cell also known as B Cell. These cells are created in the bone marrow and, when activated, are able to produce antibodies, which bind to the antigen during the immunological response as a means to destroy it. This can only happen if there is a match between the antibody and the surface of the foreign material. Humans are thought to have 10^7 to 10^8 different antibodies with distinct chemical compositions [5] that account for the possible variations of antigens that one may find during the course of a lifetime.

The second kind of adaptive response is called cellular immunity and is mediated by lymphocytes called T cells. These cells are also produced in the bone marrow, but are matured in an organ named Thymus. They are responsible for killing tumor cells and cells from the body that were infected by the pathogen (altered self-cells).

Overview of the Immune Response



Punt, *Kuby Immunology*, 8e, © 2018 W. H. Freeman and Company

Figura 1.1: Main Processes of an Immune Response [1]

Figure 1.1 shows an overview of the main activities that may happen during an immune response. The whole process starts with an infection caused by an antigen that was capable of passing through the physical and chemical barriers of the body (1). After that, when the pathogen is detected by the front line phagocytic cells, a hormone-like protein called cytokine is released by them as to induce a local inflammatory response (2). Besides that, these cells are capable of engulfing the antigen and transporting it through the lymphatic vessels with the objective of enacting the adaptive immune response (3).

In the meanwhile, both B and T lymphocytes are derived in the Bone marrow via a process called hematopoiesis. The receptors of these cells undergo a pseudo-random genetic rearrangement process during their creation that account for the variety of cells, and thus the ability to bind with unseen substances [9]. While the B cells flow directly through the blood flow into secondary lymphoid organs, the T cells stop at the Thymus to be matured, and then follow the same path as the B cells (4). The maturation of the T cells is a censoring process in which the lymphocytes are tested against proteins of the body. If a T-cell strongly binds to some self-protein, it is discarded. Hence, only the T-cells that did not have a strong bind are allowed to flow through the bloodstream and be used against the pathogen. This process is called Negative Selection and aims at avoiding autoimmune responses.

The adaptive immune response starts at a secondary lymphoid organ, with the arrival of the phagocytic cell carrying the antigen. In this step, a mechanism called Clonal Selection is performed (5). The B and T lymphocytes with a high level of engagement with the pathogen proliferate and mutate (somatic hypermutation) as a means to grow in number and to improve the affinity with the foreign substance. Afterwards, these differentiated cells leave the lymphoid organ and are pumped throughout the circulatory system by the heart (6) until reaching the local of the inflammatory response (7). Finally, the specialized lymphocytes act on the antigens in order to destroy the residual of the invasion (8). These cells are kept as memory cells so that, in the case of a future similar threat, allow for a faster response.

1.1.2 Description of the Negative Selection Algorithm

One of the fundamental skills of the BIS is the ability to differentiate between the body's own cells and the foreign material. It is called self/nonself discrimination [2] and helps to protect the body from attacking itself whilst building a strong defense against foreigners.

As described in the previous subsection, T-cells are created in the bone marrow, where they have their receptors differentiated by a pseudo-random genetic rearrangement process. Later, in the Thymus, they are maturated by being tested with self cells. Lymphocytes that have a strong binding with self cells are discarded, in a censoring fashion. The result of the process is a set of diverse T cells specialized in binding with nonself cells.

In 1994, Forrest et al. introduced an algorithm inspired by this process that is used in the field of change/anomaly detection [10]. They have abstracted the concepts of self and nonself so that the process of Negative Selection could be used in more general problems. They have used a file authentication system as an example to instantiate one of the possible applications. In their work, data from legitimate users trying to access a file were thought as the body's own cells, while non-authorized accesses were seen as nonself material, or pathogens. Both types of data were translated into binaty strings, and T-cell detectors were generated as binary strings of the same size which, during the censoring phase, did not match the self data. A change was considered whenever there was a match between some detector and the new data in the monitoring phase.

Since then, even though the main idea of the algorithm was kept, several implementations and adaptations were made to improve its performance and allow for the application it in different scenarios. Dasgupta [2], in a recent work, reviewed the literature on this matter and classified the algorithms found based on the following characteristics: data type, data representation, distance function, detector size and initialization.

- **Data Type:** Refers to the type of the data that is used by the algorithm. It can be binary or real value.
- **Data Representation:** Related to how the data is structured or formatted. If the data type is binary, it can be shaped as a string or a grid. If it is a real value, the formats are grid and vector.
- **Distance function:** Also called "affinity"or "matching"Function, it is a function that identifies how strong is the bind between the self and nonself data. In the case of binary strings, the most common are the r-chunk, r-continuous bits and hamming distance with its variations. When talking about real valued types, the functions are usually the distance between vectors, like the manhattan, euclidean or minkowski distance. This section will provide more details about this topic latter on.
- **Detector Initialization:** Describes how the detectors are generated. In both data types it can be in a random, semi random or adaptive fashion. Further details are provided below.
- **Detector Size:** It is usually fixed, but in some real valued algorithms the detectors' size may vary in size in the generation process.

Some discussion has been made over the Data Type and Representation characteristics since they limit all the others [11] [2] [8]. The advantages of using bit strings are that

(1) any data can be presented as a binary string, (2) it facilitates the analysis of the result and (3) categorical data are well represented in this form. Nevertheless, they have a scalability issue that comes with the increased string size. All in all, to achieve the goals of this work, the binary data type and the string representation will suffice.

Even though different implementations may fall in different buckets, Ji and Dasgupta [11] have distilled the three aspects that must be present in an algorithm so that it may be considered a Negative Selection Algorithm. They are:

1. The goal is to identify the self-set counterpart.

2. Change/anomaly detection is performed by using some form of detector.

3. The algorithm makes use of only the self-samples during the detector generation.

The next subsections provide a more in depth view of the algorithm for the binary data type, by showing its overall structure, detailing how the comparisons with the monitored data are made and how the detectors are generated.

Algorithm Description

The Negative Selection Algorithm (NSA) can be divided in two steps: the generation of the detectors, and the actual process of detection of the nonself. These steps are similar to most supervised algorithms, in which there is a training and a test phase [2].

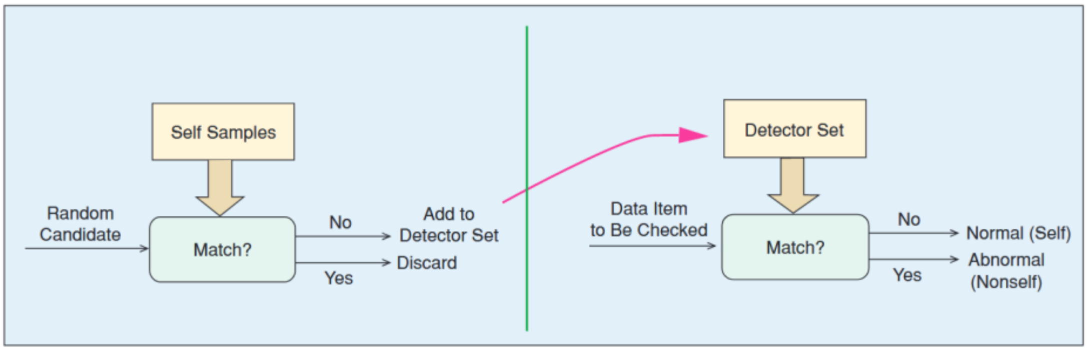


Figura 1.2: The two steps of the NSA [2]

Figure 1.2 helps to shed some light on both phases. On the left-hand side, the generation of the detectors is illustrated. In this "training"step, a set of random candidates is generated by some predefined process and undergo a censoring process based on the self samples. The candidates that match the self samples, measured by the distance function, are discarded, while the ones that do not match are added to the nonself detector set. On the right-hand side of the figure, the detection, or "test", phase is shown. The nonself detector set obtained in the first step is tested against the data that is being monitored.

The same distance function is utilized here to check whether the new data matches with some of the detectors in the set. In the case of a match, the data is considered abnormal.

For a complete understanding of the inner workings of the algorithm, two important pieces are missing: a description of how the random detector candidates are generated and the definition of the matching function that will account for the affinity measurement.

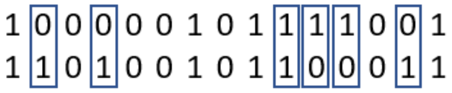
Distance Function

First, the Distance function, or matching rule, is an operation that relies on the comparison of characters (or bits) between two strings and provides a score telling how similar (or different) the strings are. Three rules are the most widely used in the literature [12]: the Hamming Distance, the R-Contiguous and the R-Chunk matching rules.

- **Hamming Distance:** This function measures the number of characters that differ between two strings [8]. Let X be a string of lenght n such that $X = x_1x_2x_3...x_n$ and let D be a dectector with the same lenght, so that $D = d_1d_2d_3...d_n$. The Hamming Distance can be formally defined as:

$$HD = \sum_i (\overline{x_i \oplus d_i})$$

  where $\oplus$ is the XOR operation. In this case, a match between X and D is said to have happened when the HD score is below a predefined threshold. Figure 1.3 shows how the comparison is made. The characters at each position are compared and, if they are different, a unit is added to the final score. Therefore, two strigs must have a low score to be considered similar.



1 0 0 0 0 0 1 0 1 1 1 1 0 0 1
1 1 0 1 0 0 1 0 1 1 0 0 0 1 1

Figura 1.3: Illustration of the Hamming Distance Matching Rule

- **R-Contiguous:** Let again X be a string of lenght n such that $X = x_1x_2x_3...x_n$ and let D be a dectector with the same lenght, so that $D = d_1d_2d_3...d_n$. Let also r be an integer, such that $0 > r >= n$. This rule defines a match between X and D whenever the two strings have at least r consecutive identical characters starting at any position. This rule was mainly used in the first implementations of the NSA, in which the detectors were created in a generate-and-test fashion [12].

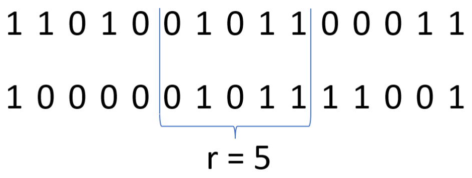


Figura 1.4: Illustration of the R-Contiguous Matching Rule

Figure 1.4 illustrates this concept. A window of size r slides searching for a region where the substrings match. If at least one region is found, then X and D are considered a match.

- **R-Chunk:** Let X be a string of lenght n such that $X = x_1x_2x_3...x_n$ and let D be a dectector of size m so that $D = d_1d_2d_3...d_m$, with $m \leq n$. Similarly to the R-Continuous rule, the string and the detector are considered a match if, at a position p, all bits of D are identical to the bits X in a window of size m, with $0 \leq p \leq n-r$. Hence, the detector is characterized by a chunk of size r and a starting position p, and can be uniquely identified as $t_{p,D}$. The practical difference between this rule and the previous one is that this function allows for detectors of any size, which improves the self-space coverage [8].

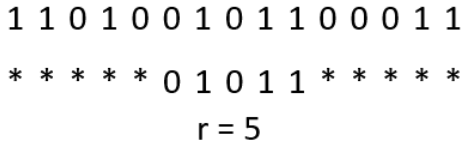


Figura 1.5: Illustration of the R-Chunk Matching Rule

Figure 1.5 shows an example of this rule. The detector $t_{6,01011}$ has size 5 and was a match in the sixth position of the string. The "*"represents irrelevant positions meaning that any character can be matched.

The R-chunk is said to enhance the accuracy and performance of the NSA [12] because the same generated string can be used as a detector in several positions. Therefore, one can say that lower sized detectors comprise an optimal detector set, since more abnormal data can be detected. Nevertheless, as cited by Wierzchon and Chmielewski [13], a study performed by Stibor showed that strings generated with low values of r are less likely to become detectors. This probability highly increases in the middle range, and is close to 1 for large string sizes. Hence, there is a sweet spot when trying to find the size of the string, which is usually for middle values of r, that aligns accuracy and coverability with efficiency.

Detector Generation

Both Ayara et al. [14] and Dasgupta and Niño [8] provide a thorough detailing of the different methods found in literature for the generation of the detectors set for binary data. The most basic approach is the exhaustive detector generation, which was introduced in the original NSA paper [10]. The idea is to exhaustively generate random candidates until a big enough set of detectors is achieved. It was reported to be very time-consuming, since the amount of candidates grows exponentially with the size of the self-set [3].

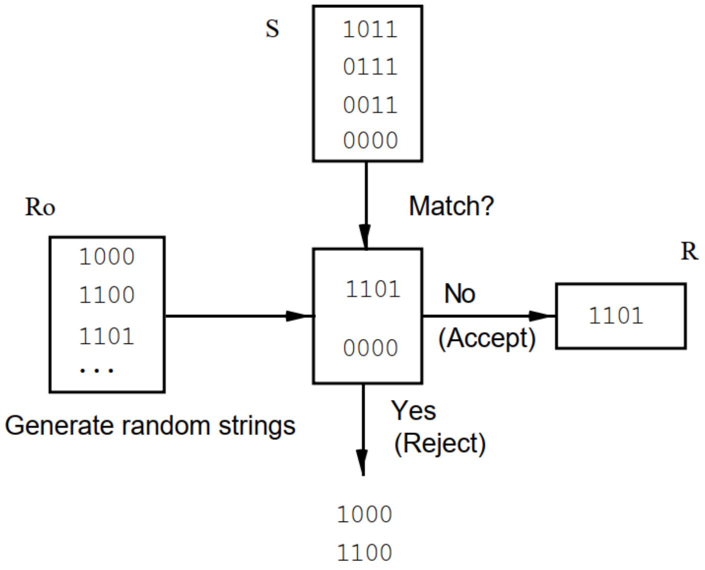


Figura 1.6: Example of the Exhaustive Detector Generation Process [3]

This generate-and-test method can be further explained with Figure 1.6 in which random binary strings are generated and compose the R_o set. Then, each candidate is tested for a match with the self-set S by using one of the functions described earlier. In the figure, if the compared strings have at least 2 matching contiguous bits, the candidate is rejected, otherwise it is accepted as a valid detector and joins set R.

The problem with the exhaustive detector generation is that a great number of candidates are rejected during the censoring, making it be inneficient [15], besides being costly in terms of computational use of resources [14]. To tackle that, two other approaches arose: the linear and the greedy algorithms, both based on the r-countinuous distance method.

The linear time algorithm is a two-phase process named after its complexity introduced by D'haeseleer et al. [15]. Initially, all the strings that are unmatched by the self-set have their recurrence counted. Then, this counting recurrence is used to naturally number the unmatched strings and allow for the picking of random detectors, according to the desired size of the detector's set. Even though this algorithm runs in linear time according to the detector's set and self-set sizes, the recurrence counting requires the storage of all the possible matches two strings can have by using the r-contiguous distance. This means that, although its complexity is linear in terms of time, it is exponential with regard to space.

The Greedy algorithm is another algorithm introduced by D'haeseleer et al. [15], which tries to provide a better coverage of the string space without increasing the amount of detectors. It does so by slightly modificating the construction of the array of possible matches. It also relies on two arrays, one that stores the candidates picked by the algorithm and other keeps track of the strings that still were not matched with any picked detector. New detectors are generated based on the unmatched strings that have the highest recurrence value [8]. This algorithm provides an optimal set of detectors but has a higher time complexity when compared with the Linear algorithm. This happens because of the update of the two arrays that happens whenever a new detector is generated. Time complexity is kept, though.

Referências

[1] Punt, Jenni, Sharon A. Stranford, Patricia P. Jones e Judith A. and Owen. W. H. Freeman Macmillan Learning, New York, eighth edition edição, 2019. vi, 2, 3

[2] Gupta, Kishor Datta e Dipankar Dasgupta: *Negative selection algorithm research and applications in the last decade: A review.* IEEE Transactions on Artificial Intelligence, PP:1–1, setembro 2021. vi, 1, 4, 5, 6

[3] D'haeseleer, Patrik: *An immunological approach to change detection: Theoretical results.* páginas 18–26, julho 1996, ISBN 0-8186-7522-5. vi, 9

[4] KamalMishra, Prashant e Mamta Bhusry: *Artificial immune system: State of the art approach.* International Journal of Computer Applications, 120:25–32, junho 2015. 1

[5] Farmer, J.Doyne, Norman Packard e Alan Perelson: *The immune system, adaptation, and machine learning.* Volume 22, outubro 1986. 1, 3

[6] Garrett, Simon: *How do we evaluate artificial immune systems?* Evolutionary computation, 13:145–77, fevereiro 2005. 1

[7] Naqvi, Syed Moeen, Merve Astekin, Sehrish Malik e Leon Moonen: *Adaptive immunity for software: Towards autonomous self-healing systems*, janeiro 2021. 1

[8] Dasgupta, Dipankar e Luis Fernando Nino: *Immunological computation: Theory and application.* Ingeniería e Investigación, 29:140–140, abril 2009. 2, 5, 7, 8, 9, 10

[9] Aickelin, Uwe, Dipankar Dasgupta e Feng Gu: *Artificial immune systems*, páginas 187–212. janeiro 2014. 4

[10] Forrest, Stephanie, Alan Perelson, Lawrence Allen e Rajesh Cherukuri: *Self-nonself discrimination in a computer.* Proceedings of the International Symposium on Security and Privacy, novembro 1995. 5, 9

[11] Ji, Zhou e Dipankar Dasgupta: *Revisiting negative selection algorithms.* Evolutionary computation, 15:223–51, fevereiro 2007. 5, 6

[12] González, Fabio, Dipankar Dasgupta e Jonatan Gomez: *The effect of binary matching rules in negative selection.* Volume 2723, páginas 195–206, julho 2003. 7, 8

[13] Chmielewski, Andrzej e Slawomir Wierzchon: *Hybrid negative selection approach for anomaly detection.* setembro 2012, ISBN 978-3-642-33259-3. 8

[14] Ayara, M., Jon Timmis, R. Lemos, Leandro De Castro e R. Duncan: *Negative selection: How to generate detectors*. Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS), janeiro 2002. 9

[15] D'haeseleer, Patrik, Stephanie Forrest e Paul Helman: *An immunological approach to change detection: algorithms, analysis and implications*. páginas 110–119, maio 1996. 9, 10