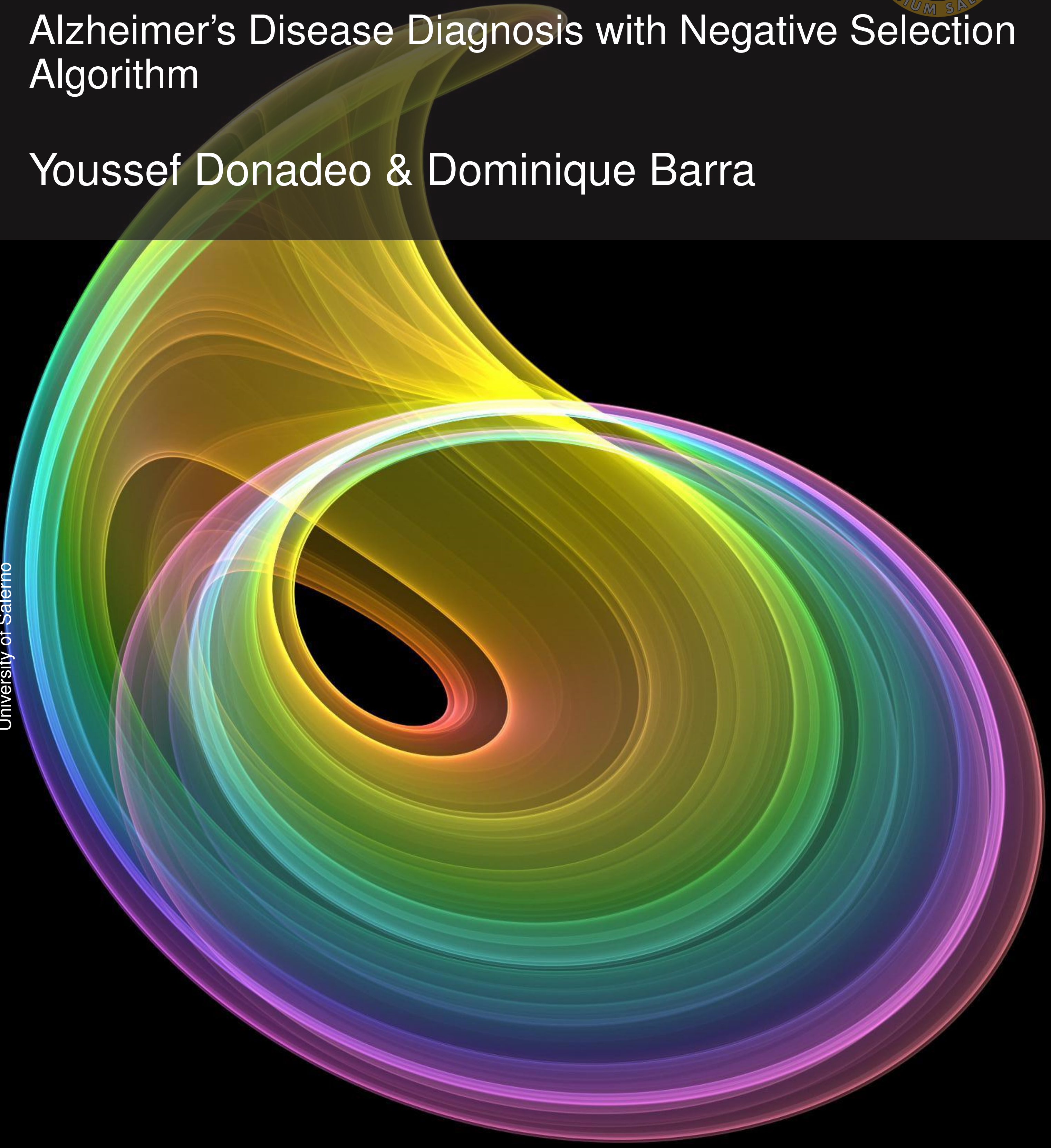
# Final Project Report

## Alzheimer’s Disease Diagnosis with Negative Selection Algorithm

Youssef Donadeo & Dominique Barra



University of Salerno

# Final Project Report

Alzheimer's Disease Diagnosis with Negative Selection Algorithm

by

Youssef Donadeo & Dominique Barra

Instructors: Prof. Angelo Marcelli, Prof. Antonio Della Cioppa

Institution: NCLab, DIEM, University of Salerno

# Contents

# List of Algorithms

# List of Tables

# 1 Introduction

Neurodegenerative diseases (NDs) affect millions of people worldwide, and among them, Alzheimer's disease is the most frequent one. It is the most common cause of dementia in people over the age of 65, and since it is age-related, the increment of the lifespan due to the continuous improvement in the medical field will make its impact on both individuals and society even greater. According to the World Alzheimer Report released by the Alzheimer's Disease International, in 2015 it affected more than 40 million people worldwide, and it is estimated this number will increase to 131.5 million by 2050 [5].

Recently, more and more researchers have focused on Artificial Immune System (AIS), which is inspired by biological immune system [1]. In AIS, Negative Selection Algorithm (NSA) is an important detector-generation algorithm, which simulates the immune tolerance in T-cell maturation process of biological immune system, and achieves effective recognition of non-self antigens by clearing self-reactive candidate detectors. Due to its robustness and self-adaption, NSA is widely used in the field of anomaly detection [3], fault diagnosis and etc. It's important to know that, from now on, in this document we will improperly refer to the Real-Valued version of the NSA (namely RNSA) just by using the acronym 'NSA' for simplicity.

Starting from the previous work done by our professors and their results on this subject [2], we started to think on how to further improve their method. NSA presents, in fact, some important drawbacks and a lot of researches were done in the years to minimizing or eliminate these [6]. Looking the data and how they are positioned in the space we find out that a possible way of further improve the performance of the NSA was to try avoiding (or minimizing) the overlaps between the found detectors and/or to have a variable radius for them. We will report the results of the previous work on Chap. 2 and we will discuss about our solutions and improvements in Chap. 3. In Chap. 4 you will find the experiments we performed to evaluate our system and the comparison with the classic NSA and, finally, conclusion and possibles further improvements are discussed in Chap. 5.

# 2 Related Work

As regards the NSA applications for supporting the diagnosis of Alzheimer's disease, we started by looking in the works [2, 4]. In this papers the authors try to overcome the problem of data collection, which we believe represents one of the major obstacles to adopting machine learning methods to develop automatic tools for this kind of disease. That's the reason why they decided to use NSA, because it results appealing and it takes only samples of one class for learning a classification model, meaning that in our case, it can be trained by using only samples produced by healthy subjects (cutting down time and effort of doing it with patients). The most remarkable result is that the proposed method achieves a 100% sensitivity on every run with a relative small dataset and an accettable number of detectors.

In the work reported in [6], the authors talks about variants of the NSA, and show the advantages and disadvantages of both the constant radius and variable radius implementation. From this document it can be highlighted that the major disadvantage for both implementations is the overlapping of the detector areas.

The variable detector radius implementation of NSA previously mentioned is called V-detector [7] and is adopted to randomly generate the sets of detectors with the aim of maximizing the coverage area of each one of them. With [8], the authors try to further improve this coverage by using an hypothesis test to understand if the detectors created are enough to cover the space of non-self samples. According to the previous paper, a statistical approach leads to the most reliable Negative Selection Algorithm. Other advantages that can be highlighted are that V-detector scheme is more effective in using smaller number of detectors because of their variable sizes, so the holes between detectors and self samples should be better covered.

# 3
# The Proposed Method

In this chapter we will firstly report, for reference, the Canonical Negative Selection Algorithm and, after that, we will present in detail the solutions to the two identified drawbacks.

## 3.1. Canonical Negative Selection Algorithm

**Algorithm 1:** NSA

```
Data: the set $S$ of $m$ self-samples in $R^n$ space, the number of detectors $N$ and the self-radius $\sigma$
Output: the detector set $D$
1:
2:  Function NSA($S$, $N$, $\sigma$):
3:      $D = \emptyset$;
4:      repeat
5:          Create a new detector as a random vector $x$ drawn from $[0, 1]^n$;
6:          foreach $s_i \in S$, $i = 1, 2, ..., m$ do
7:              $d_i = Distance(x, s_i)$;
8:              if $d_i > \sigma$ for all $i$ then
9:                  Add $x$ to $D$;
10:     until $D$ contains $N$ valid detectors;
11:     return $D$;
```

This is a simple and effective algorithm but has some important drawbacks and, among them, one that can be quickly identified: there is never a check (or some sort of *selection*) in the code to avoid that two or more detectors **overlap** between each other. This certainly allow the algorithm to reach convergence very quick, but also means that, especially if the chosen number of detectors $N$ is very big, it will generate a lot of useless detectors.
Furthermore, another important drawback is that this algorithm allows only to select a fixed radius as input and nothing more. This will force you to decide whether to select a big radius and a small number of detectors to remove big portions of the space (without being able to cover the small spaces between the sample though) or to select a small radius but a very big number of detectors, or a compromise between. That's why, another solution to improve performance would be to use a **variable radius**.
We developed and propose the solutions of both these problems here in the next sections.

## 3.2. Negative Selection Algorithm with Overlap Management

**Algorithm 2:** NSA with Overlap Management

```
Data: the set $S$ of $m$ self-samples in $R^n$ space, the number of detectors $N$, the self-radius $\sigma$,
      the amount of intersection tolerated $I \in [0, 1]$, the tolerated intersection increment $\Delta I$ and
      the patience $p$
Output: the detector set $D$

Function NSA($S$, $N$, $\sigma$, $I$, $\Delta I$, $p$):
    $D = \emptyset$;
    $f = 0$ ;                                                  /* The fatigue amount */
    repeat
        Create a new detector as a random vector $x$ drawn from $[0, 1]^n$;
        foreach $s_i \in S$, $i = 1, 2, ..., m$ do
            $d_i = Distance(x, s_i)$;
            if $d_i > \sigma$ for all $i$ then
                if CheckIntersection($x, D, I, p, f$) then
                    Add $x$ to $D$;
    until $D$ contains $N$ valid detectors;
    return $D$;

Function CheckIntersection($x$, $D$, $I$, $p$, $f$):
    if $I = 1$ then
        return true;
    foreach $x_j \in D$, $j = 0, 1, ..., len(D)$ do
        $i_j = Intersection(x, x_j)$;
        if $i_j \leq I$ for all $j$ then
            $f = 0$;
            return true;
        else
            $f = f + 1$;
            if $f > p$ then
                $f = 0$;
                $I = \min(1, I + \Delta I)$;
            return false;
```

This is our first solution. We take the canonical NSA and simply add a very naive check on the detectors overlap. Doing that, we introduced three new hyper-parameters:

- $I$ - the amount of intersection tolerated between two detectors. This is a value in the range $[0, 1]$ representing a percentage (e.g $0.6$ would mean that only detectors which share at most $60\%$ of the same space will be accepted);
- $\Delta I$ - the intersection increment. This is needed, together with patience, to avoid being stuck in an endless loop if there is no way of finding a detector with an amount of intersection above $I$;
- $p$ - the amount of patience. This is a simple integer value that is used in conjunction with a counter ($f$) and act as a threshold.

Their usage is the following: whenever the system finds a bad detector (i.e. intersection above $I$) the counter is incremented by $1$ and its value is compared with the value of $p$, if the counter exceed this threshold, $I$ will be incremented by $\Delta I$ and the counter reset. Instead, when the system find a good detector, only the counter reset will be performed.

It's also easy to see that this is a generalization of the Canonical Negative Selection Algorithm, indeed, if you set $I = 1$ you will have exactly the same algorithm.

## 3.3. Variable Radius Negative Selection Algorithm

**Algorithm 3:** VRNSA

```
Data: the set $S$ of $m$ self-samples in $R^n$ space, the number of detectors $N$, the self-radius $\sigma$,
      the percentage of the non-self area you want to cover $p$ and confidence levels $\alpha$
Output: the detector set $D$
1:
2:  Function VRNSA($S$, $N$, $\sigma$, $p$, $\alpha$):
3:      Create $n$ with value greater than $\max[\frac{5}{p}, \frac{5}{1-p}]$;
4:      $D = \emptyset$;
5:      $N = 0, x = 0$;
6:      repeat
7:          Create a new detector as a random vector $a$ drawn from $[0,1]^n$;
8:          $detector\_radius = inf$;
9:          foreach $s_i \in S$, $i = 1, 2, ..., m$ do
10:             $d_i = Distance(a, s_i)$;
11:             if $d_i > \sigma$ then
12:                 if $d_i - \sigma <= detector\_radius$ then
13:                     $detector\_radius = d_i - \sigma$;
14:             else
15:                 Goto line 7;
16:         $N = N + 1$;
17:         foreach $detect_i \in D$ do
18:             $d_i = Distance(a, detect_i)$;
19:             if $d_i < radius\ detect_i$ then
20:                 $x = x + 1$;
21:                 $z = \frac{x}{\sqrt{n*p*(1-p)}} - \sqrt{\frac{p*n}{1-p}}$;
22:                 if $z > z_\alpha$ then
23:                     Goto line $29$
24:                 Goto line $7$
25:         Save new detector $a$ and the relative radius $detector\_radius$;
26:         if $N = n$ then
27:             $N = 0, x = 0$;
28:     until $D$ contains $N$ valid detectors;
29:     return $D$;
```

The above algorithm was created on the works illustrated at [7, 8]. This is a more complex version of the one in Section 3.1 but also very effective. In this algorithm we use the coverage estimation to early stop the execution and avoid unnecessary detectors. The estimation is done using "The Hypothesis Test" where the null hypothesis is "The coverage of the non-self region by all the existing detectors is below percentage $p_{min}$". If we accept the null hypothesis, we will include more detectors. To implement this solution we introduced two new hyper-parameters:

- $p$ - Percentage of non-self area we would like to cover;
- $\alpha$ - Confidence level, It is the maximum acceptable probability that we make a Type I error (to make a false positive conclusion).

As is canonical counterpart, also this algorithm has some important drawbacks and, among them, we highlight the partial **overlapping** of detectors.

# 4 Experimental Results

## 4.1. Dataset

The Dataset we used is the same one proposed in [4]. It includes 25 different tasks, grouped into four categories, in ascending order of difficulty:

- Graphic tasks: to test the ability to write elementary moves, connect dots, and draw simple figures;
- Copy and Reverse Copy tasks: aimed at evaluating the ability to reproduce complex graphic gestures, such as letters, words, and numbers;
- Memory tasks: the purpose is to test the variation of the graphic section, keeping in memory a word, a letter, a graphic gesture, or a motor plan.
- Dictation: to test the patient's ability to use the working memory during the variation of writing tasks.

The features indicated in the table below were then detected from these tasks.

**Feature List**

| **Feature** | **Name** | **Description** |
|---|---|---|
| Total Time | TT | Total time required to perform the entire task |
| Air Time | AT | Time spent near the sheet, with the tip of the pen not in contact with the sheet |
| Paper Time | PT | Time spent on the sheet, with the tip of the pen in contact with the sheet |
| Mean Speed on-paper | MSP | Average speed recorded on paper |
| Mean Speed in-air | MSA | Average of the speeds recorded near the sheet |
| Mean Acceleration on-paper | MAP | Average of accelerations recorded on paper |
| Mean Acceleration in-air | MAA | Average of the accelerations recorded near the sheet |
| Mean Jerk on-paper | MJP | Average of the jerk recorded on the sheet |
| Mean Jerk in-air | MJA | Average of the jerk recorded near the sheet |
| Pressure Mean | PM | Average of the pressure levels exerted on the sheet |
| Pressure Var | PV | Variance of pressure levels exerted on the sheet |

| | | |
|---|---|---|
| GMRT on-paper | GMRTP | Generalization of the Mean Relative Tremor defined in computed on in-air movements. It considers the top left corner of the sheet as the center for the computation |
| GMRT in-air | GMRTA | Generalization of the Mean Relative Tremor defined in computed on on-paper movements. It considers the top left corner of the sheet as the center for the computation |
| Mean GMRT | GMRT | Average of GMRTP and GMRTA |
| Pendowns Number | PWN | The number of pendowns |
| Max X Extension | XE | Maximum extension recorded along the X axis |
| Max Y Extension | YE | Maximum extension recorded along the Y axis |
| Dispersion Index | DI | It measures the dispersion of the drawing on the sheet; a fully covered sheet will correspond to an index equal to one while a completely empty sheet will have an index equal to zero |

**Table 4.1:** Features used for the experiment.

The data collection campaign involved 174 individuals, of which 89 are Alzheimer's patients, and 85 are healthy subjects. All subjects were recruited so as to match the patients and the control group for age, education level, type of occupation, and gender. The two populations show similar mean values and standard deviations for age, sex, and years of education. Each of these samples is represented by a feature vector of 450 real values. A reduced dataset including 107 features for each sample was obtained by applying the *Searching for Uncorrelated List of Variables (SULOV)* feature selection method.

## 4.2. Implementation Details

The experiments were conducted using the Canonical Negative Selection Algorithm and the previous results reached [2] as base.

As regarding the Negative Selection Algorithm with Overlap Management, we firstly tried to use the same settings used before on NSA ($n = 4000, \sigma = 5.03$) and then we tried to reduce $n$ (at $3000$, $2500$ and $1500$) to see if our algorithm would bring some improvements or not. In all the tries $I$ was set equal to $0.6$ and $\Delta I$ to $0.05$. This choice come from some tests we performed on the dataset: essentially, our data are enough close together, whenever we chose a $I$ less then this one, with the $\sigma$ we selected for our experiments, there was not enough space to find any more detectors then a couple ones and, due to this, the model would have taken a very long time for training.

For Variable Radius Negative Selection Algorithm, instead, this approach wasn't feasible because the radius we are going to set refers to that of the self samples and the one of the detectors is computed automatically for each one of them. Anyway, we have always considered the upper limit of the detectors radius to be used $n = 4000$ and changed more self radius to see the possible improvements($\sigma \in [0.05, 0.02]$). In the end, we have chosen ($n = 4000, \sigma = 0.03, p = 0.999, \alpha = 0.025$).

## 4.3. Performance Evaluation

The training was performed using $80\%$ of the healthy subjects samples for the training set and the remaining $20\%$ along with $100\%$ of the patients samples for the test set. Moreover, to perform a fair comparison, we have performed 15 runs by selecting at random the samples of the healthy subjects to be included in the training set.

Table 4.2 reports the mean, the standard deviation, the maximum, and the minimum of the accuracy achieved across the 15 runs between the different classifiers. All three algorithms results were tested with a Mann-Whitney U test, from it we know that the Canonical NSA and its version with Overlap Management follow the same distribution, but this is not true for the VRNSA so could not be directly compared only with these simple statistics and some others studies should be done to really understand

| | NSA | NSA with Overlap Management | | | | VRNSA |
|---|---|---|---|---|---|---|
| *Nr. Detectors* | 4000 | 1500 | 2500 | 3000 | 4000 | 5-11 |
| **Average accuracy** | 96,54% | 97,36% | 96,89% | 96,35% | 95,03% | 89,28% |
| **Standard deviation** | 2,14% | 2,04% | 1,89% | 2,14% | 2,7% | 5,93% |
| **Max accuracy** | 100% | 100% | 99,05% | 100% | 100% | 100% |
| **Min accuracy** | 91,51% | 93,39% | 92,45% | 90,57% | 89,62% | 66,03% |

**Table 4.2:** Average accuracy, standard deviation, maximum and minimum obtained between the algorithms in 15 runs.

| | NSA | NSA with Overlap Management | | | | VRNSA |
|---|---|---|---|---|---|---|
| *Nr. Detectors* | 4000 | 1500 | 2500 | 3000 | 4000 | 5-11 |
| **Sensitivity** | 99,85% | 97,90% | 99,55% | 99,72% | 100% | 96,38% |
| **Specificity** | 96,24% | 99,00% | 96,91% | 96,16% | 94,50% | 92,05% |

**Table 4.3:** Average sensitivity and specificity in 15 runs.

if its results should be considered better or not of the other ones. Anyway, what can we conclude from the data in the table is that all the models perform particularly well on the test set and we can clearly see the improvements that the Overlap Management has: essentially, with less than an half of the detectors of the Canonical version, we obtained almost the same results and, considering that $\Delta I$ never went above $0.65$ in any of the runs, we have potentially covered more space.

It is worth notice that VRNSA has a higher standard deviation than the others, so on 15 runs it has accuracy values distributed over a wider range. This is probably because every run generate a very small number of detector that, statistically, should be enough to cover all the space but, practically, are not sufficient.

Another important result, follow from Table 4.3 that reports a very high sensitivity and specificity for all models. This is a very important result from [2] and that is kept intact in our implementations. In particularly, a value of sensitivity very high, like the ones here reported, means that is very likely that, almost every time the system says someone does not suffer from the disease, it will be right.

The table also shows that VRNSA, despite having lower sensitivity and specificity, obtained values very close to the other classifiers using a significantly smaller number of detectors.

From Table 4.4 we can point out that VRNSA has a lower number of generated detectors due to the estimate of the coverage of non-self samples. The accepted detectors are approximately 1% of those generated for all the models, because many of the detectors generated after the accepted ones fell within the area covered by the latter.

Furthermore we can see, in the management of the overlap, as the desired detectors increase, the number of detectors generated and rejected increases too (and a lot). This is clearly due to the check on intersection that discard a major portion of them, especially in a problem like this, where the samples are very close together and find two detector that don't overlap and can, at the same time, cover all the space it's a difficult task.

| | NSA | NSA with Overlap Management | | | | VRNSA |
|---|---|---|---|---|---|---|
| *Nr. Detectors* | 4000 | 1500 | 2500 | 3000 | 4000 | 5-11 |
| **Detectors Generated** | 22431949 | 23901113 | 34735804 | 40743976 | 53919988 | 5330 |
| **Detectors Discarded** | 22427949 | 23899613 | 34733304 | 40740976 | 53915988 | 5322 |

**Table 4.4:** Average of detectors generated and discarded in 45 runs.

# 5 Conclusions

The paper proposed a classification study of patients suffering from Alzheimer's disease using as classification method the "Negative Selection Algorithm". This method was chosen because it turns out to be non-invasive, easy to perform, requires only an inexpensive device to collect data and the software can be installed on small IT devices, such as a tablet, thus making the whole system easy carry on (as mentioned in [2]).

In this study we tried to overcome some problems that the algorithm presents such as:

1. The overlapping of the detectors
2. The curse of dimensionality

To try to overcome these problems we used two algorithms, Negative Selection Algorithm with Overlap Management and Variable Radius Negative Selection Algorithm, proving that it's possible to obtain comparable performance in terms of accuracy, sensitivity and specificity, using less detectors and reducing time complexity (this last one only in VRNSA, thanks to the coverage check) (Section 4.3).

Due to the relatively long training time, we have not been able to verify other values combination for the parameters of the Negative Selection Algorithm with Overlap Management, therefore, in future studies, it could be a good idea to explore them and see how would performs with different radius and amount of tolerated intersection. Another important thing noticed is that, with the same amount of detectors of the canonical algorithm, our implementation can reach nearly the same performance and coverage of the current data: this means that probably, since this version tends to keep the detectors away from one another, it is able to cover more of the space distant from the *self* samples (or the empty space in general). It could be a very important result, shared also from the VRNSA version, because this would mean that the model is more robust and resistant to the introduction or changing of data.

In our opinion, the most remarkable result is that the VRNSA method achieves, on average, a good sensitivity value with a number of detector in a range $[5, 11]$ on every run. Sadly, the distribution of the results didn't pass the hypothesis test, and we weren't able to find another way to statistically compare the results. Despite this, we have studied the dataset trough experiments and PCA reduction (at two and three dimension. The reduction, although decreased the variance of the data, was necessary because it would be impossible to plot or understand anything about them due to their high number of feature) and find out that the samples are very close together and the space surrounding them is quite wide. In these conditions, the variable radius of the VRNSA should be the best solution, at least intuitively, because it will remove a big portion of the empty space with only 2-3 detectors with big radius and then try to cover the remaining space between samples with smaller detectors.

# References

[1] S. Yu D. Dasgupta and F. Nino. *Recent advances in articial immune systems*. Appl. Soft Comput, 2011.

[2] Antonio Della Cioppa Giuseppe De Gregorio and Angelo Marcelli. *Negative Selection Algorithm for Alzheimer's Diagnosis: Design and Performance Evaluation*. Via Giovanni Paolo II 132, 84084 Fisciano, SA, Italy: University of Salerno.

[3] L.F.Chen. *An improved negative selection approach for anomaly detection: With applications in medical diagnosis and quality inspection*. Neural Comput. Appl., 2013.

[4] Angelo Marcelli Nicole D. Cilia Giuseppe De Gregorio, Francesco Fontanella, and Antonio Parziale. *Diagnosing Alzheimer's disease from on-line handwriting: A novel dataset and performance benchmarking*. Elsevier Ltd, 2022.

[5] Martin James Prince et al. *World Alzheimer Report 2015: The Global Impact of Dementia.* London, UK, 2015.

[6] Chikh Ramdane and Salim Chikhi. *Negative Selection Algorithm: Recent Improvements and Its Application in Intrusion Detection System*. Constantine, Algeria: Constantine 2 University, 2017.

[7] Dipankar Dasgupta Zhou Ji. *Real-Valued Negative Selection Algorithm with Variable-Sized Detectors*. The University of Memphis Memphis, 2004.

[8] Dipankar Dasgupta Zhou Ji. *V-detector: An efficient negative selection algorithm with "probably adequate" detector coverage*. The University of Memphis Memphis, 2008.