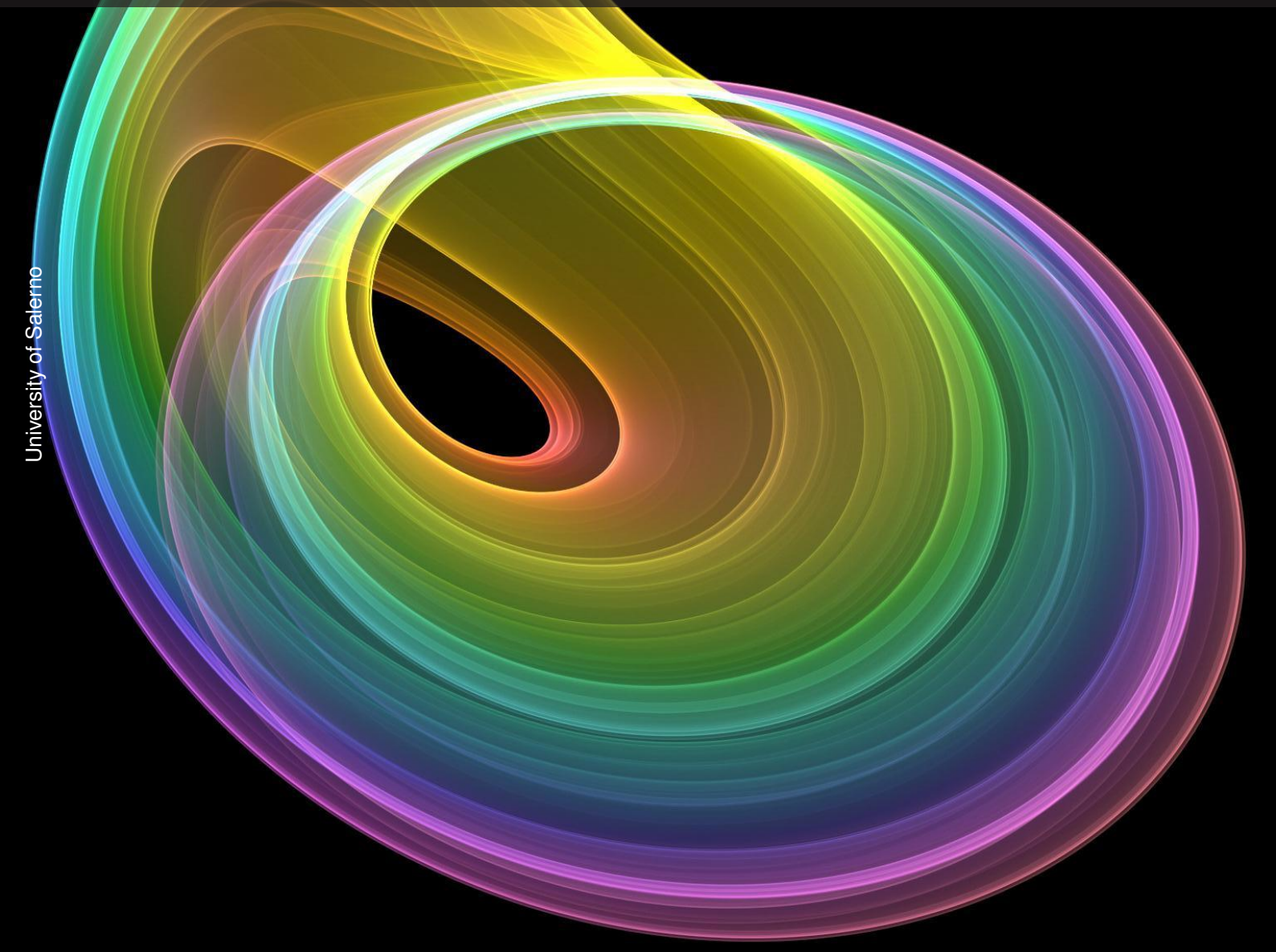
Final Project Report

Project: Genetic Programming for Alzheimer's Diagnosis

Course: Natural Computation

Alfonso Cavallo
Antonietta Napoli
Simona Sorgente

University of Salerno



Final Project Report

Project: Genetic Programming for Alzheimer's Diagnosis

by

Alfonso Cavallo
Antonietta Napoli
Simona Sorgente

Student Name	Student Number
Alfonso Cavallo	0622701441
Antonietta Napoli	0622701515
Simona Sorgente	0622701481

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

Institution: NCLab, DIEM, University of Salerno

Abstract

This work explores a method to diagnose Alzheimer's Disease starting from handwritten data analysis. Taking into account that state-of-art is mostly focused on Machine Learning approaches, we provide an alternative based on Genetic Programming. Its main advantage is to produce interpretable solutions which are majorly demanded in medical field. Experimental results, conducted on 174 subjects, proved that this method, despite it being less performing than well-known solutions, is a promising way to build explicit models to support diagnosis. Such models can provide insights for the design of cheap, non-invasive and easy to administer protocols.

Contents

Abstract i

List of Figures iii

List of Tables iv

1 Chapter 1: Introduction 1

2 Chapter 2: State of the art 2
2.1 Related Work . . . 2

3 Chapter 3: Proposed Method 4
3.1 Genetic Programming . . . 4

4 Chapter 4: Experimental Results 6
4.1 Dataset Analysis . . . 6
4.2 Experimental Design . . . 6
4.3 Implementation Details . . . 7
4.4 Tuning Phase Results . . . 7
4.5 Feature Selection . . . 7
4.6 Performance Evaluation . . . 9

5 Conclusions 12

References 13

List of Figures

1 Designed workflow for the experiment 7
2 Distribution on RD and RDAS 10

List of Tables

1 Chosen Hyperparameters 8
2 Function Set 8
3 Performance comparison between Complete Dataset (CD) and Reduced Dataset (RD) . 8
4 Details of Features Selected (in decreasing order of importance) 9
5 Final Results 10
6 Performance of GP on Accuracy, Sensitivity and Specificity got from Training Set (TRN) and Test Set (TST) 10

1

Chapter 1: Introduction

Alzheimer's is the most common age-related neurodegenerative disease (NDs). According to the International Alzheimer's Committee, more than 55 million people worldwide suffer from Alzheimer's today and it is estimated that by 2050 the number will rise to 15 million [17]. This disease is due to the degeneration of the brain tissues which causes the death of neurons: the connections between neuronal networks can dissolve and many regions of the brain become smaller and smaller. In the last stage of this pathology, this process called *atrophy* spreads a lot and leads to a significant loss of brain volume. As a result, motor and cognitive functions will be affected, and, as it worsens over time, making movements becomes a challenging action for sufferers. In addition, other major symptoms will be memory loss and language impairment. There are no clear causes: the cause could be a combination of factors related to genetics, environment, and age.

Nowadays there are still no known treatments that can stop or slow the progression of Alzheimer's, hence an early diagnosis is crucial. To improve the diagnostic workup of NDs, facilitate the development, and monitor the effectiveness of disease-modifying therapies, using biomarkers is one path to choose. By definition, a biomarker is "*a characteristic that is objectively measured and evaluated as an indicator of normal biological processes, pathogenic processes, or pharmacological responses to a therapeutic intervention*" [12]. However, these biomarkers present some disadvantages: they are expensive, invasive, and may require a long time to be administered. Moreover, one possible method to overcome these problems is to use *handwriting analysis* to both diagnose the NDs and evaluate their progression. In the recent decades, indeed, researchers have discovered the efficiency of handwriting analysis since the handwriting of Alzheimer's patients shows alterations compared to the writing of a healthy subject due to the symptoms of the pathology.

According to state of the art, Deep Learning (DL) is used to diagnose this disease but these approaches require a significant amount of data which is difficult to collect due to privacy-related issues. Moreover, another disadvantage of DL is their lack of interpretability. To face the problems we have currently discussed, we introduce an algorithm based on Genetic Programming (GP) to classify healthy and ill subjects since GP returns an explicit solution that can be seen as a function. In the medical field, since the interpretability of an automatic system is mandatory, knowing how to achieve the final solution is crucial for the specialists in the sector. To evaluate the performance of the proposed approach, the dataset collected within the DARWIN project has been used.

The paper is organized as follows. In this first chapter, we give an overview of the problem of search. In Chapter 2, we give a review of state of the art in order to observe the performance of the systems that regards Alzheimer's Disease (AD) in literature. Chapter 3 summarizes the main features of the GP algorithm. Chapter 4 illustrates the implemented solution, describes the experiments we performed to evaluate the effectiveness of the proposed system and reports the obtained results. Finally, conclusions and future research directions are discussed in Chapter 5.

2

Chapter 2: State of the art

In this chapter, we explore the works related to handwriting analysis for the diagnosis of Alzheimer's Disease and other neurodegenerative pathologies through evolutionary methods. A quick overview of the papers that influenced our study along with other articles concerning the same topics is provided.

2.1. Related Work

According to [9], many different solutions to problems related to Alzheimer's diagnosis through handwriting analysis were tested. The solution presented in [9] uses two datasets named as follows: *Complete Dataset* (CD) and *Reduced Dataset* (RD). The approach presented in that paper is the Negative Selection Algorithm (NSA) whose advantage is to produce a binary classifier, exploiting only healthy subject data to select the detectors. Performance on CD reaches 89.58% of average accuracy, 100% of specificity, and 83.14% of sensitivity. Performance on RD reaches 96.58% of average accuracy, 86.67% of specificity, and 100% of sensitivity. The solution presented in our paper explores the same problem of [9] evaluated on RD. However, the approach used to obtain the results, in our case, is different from the paper just mentioned. Before going into the details of the experimental results of our solution, a brief description of state-of-the-art works based on handwriting analysis to discriminate healthy subjects from patients is reported.

In [18], one of the first attempts to look at the handwriting process of subjects for discriminating patients affected by Mild Cognitive Impairment (MCI) from healthy subjects is described. In this study, the handwriting process has been kinematically evaluated through different functional tasks that can be exploited. Data has been collected taking into consideration 31 individuals with MCI, 22 with Alzheimer and 41 healthy subjects. After the experimentation phase, it has been achieved an accuracy between 69% and 72%.

In [16] a new approach for early diagnosis of NDs evaluating the dynamic of handwritten signatures has been presented. A sigma-lognormal model has been applied taking into consideration 12 features extracted from the signatures. *Classification And Regression Trees*(CART), bagging CART, and *Support Vector Machine* (SVM) have been used for the classification of signatures. The bagging CART model shows the best performance with an accuracy of 96.7%, with sensitivity and specificity equal to 96.5% and 96.8% respectively.

The aim of [8] is *"to compare the kinematic characteristics of handwriting and drawing between patients with AD, patients with mild cognitive impairment (MCI) and healthy controls"*. To evaluate the performance, samples from 52 participants (23 AD, 12 MCI, and 17 healthy controls) have been collected and the results have shown an accuracy ranging between 63.5% and 100%.

The work reported in [11] analyses online handwritten cursive loops in early-stage Alzheimer's patients by modeling the loop velocity trajectory with an unsupervised method. In addition, the loops have been exploited through a temporal clustering based on K-medoids. Then, for classification, has been considered an aggregation of clusters by probabilistically combining the discriminative power of each. Data have been collected from 27 early-stage Alzheimer's patients and 27 healthy subjects. The reached performance is 74% of accuracy, 72.2% of specificity, and 75.6% of sensitivity.

In [4] researchers have shown that patients affected by AD exhibit alterations in spatial organization and poor control of movement. For the purpose of the study, an experimental protocol has been used that includes tasks such as the copy of words, letters, and sentences. Two different classifiers were trained: *Random Forest* (RF) and *Decision Tree* (DT). The classifiers achieved an overall score of 70%. Afterward, performance increased using *Artificial Neural Networks* (ANN) and SVM [5].

In study [10], authors extracts features from five representative tasks related to NDs and collected samples from 71 seniors. The model has reached a final accuracy of 74.6%.

Regarding Parkinson's disease (PD) diagnosis through handwriting analysis, in [14] took into consideration DT, RF, and SVM. For the performance evaluation, a public dataset has been exploited, and the results show that the system based on the DT achieves comparable or better results than state-of-the-art solutions.

Furthermore, in [15] it has been used *Cartesian Genetic Programming* (CGP) as a classification method showing an improvement in performance in terms of both accuracy and interpretability.

At the best of our knowledge there is only one work that addresses the problem of distinguishing Alzheimer's patients from healthy ones using GP [1]. In this paper, it is presented an approach based on *Vectorial Genetic Programming* (VGP) to recognize the handwriting of AD patients. This approach was employed to analyze vectorial features such as time series. In this case, the dataset was composed of raw coordinates of the pen during the execution of the handwriting task. The average performance on the accuracy, for each of the nine tasks, is included in the range 66.15% and 76.80%.

3

Chapter 3: Proposed Method

In this chapter, the general concepts of Genetic Programming are explored.

3.1. Genetic Programming

According to literature, Genetic Programming (GP) has a significant and wide application field. For instance, the GP algorithm is employed for feature construction, learning, and ensembles for image classification [13]. The idea originated in the 1950s by Alan Turing but was developed in the USA, in the 1990s, by John Koza [6]. This algorithm uses the paradigm of Evolutionary Computation to breed computer programs that can perform typical Machine Learning tasks. GP competes with Neural Networks and, like it, needs huge populations (thousands of individuals). This is the reason why this approach is computationally very expensive, but, with the increasing CPU powers, it slowly also becomes applicable for complex problems.

The structure used for the evolution is a tree structure (hence we are not taking into consideration linear chromosomes differently from many other methods) because it's a good representation of both pc programs, functions, and expressions. In this sense, GP is an evolutionary approach that extends the genetic algorithm paradigm by allowing the exploration of the space of computer programs. In fact, in GP algorithms *exploration* is not conducted in a binary solution space, therefore, the task is more complex since we have an infinite number of programs that can solve the problem and this means that the search space is not limited.

GP defines a quality criterion (or fitness) and then uses it to evolve a population of candidate individuals by mimicking the Darwin evolution's theory. To breed the solutions to the problems, it uses an iterative process involving a probabilistic or deterministic selection of the fittest solutions and their variation employing a set of genetic operators such as the typical crossover and mutation. Each solution is a symbolic expression composed of elements from a terminal set and a non-terminal set (also known as Function Set). The terminal set includes variables and constants of the problem and their values; the non-terminal set includes functions and operations that are allowed on the features. Such a solution can be represented as a tree where internal nodes are operations of the non-terminal set and leaves are terminal elements. While defining the representation in GP, syntactic closure and sufficiency have to be taken into account [6]. An exhaustive review of the GP is reported in [6]. The process used by GP is described by the Algorithm 1.

The *initialization* is performed by firstly choosing a maximum and a minimum depth for the tree. Afterward, the initial population is randomly generated according to those parameters. Typically, the parent selection is based on fitness. In fact, a *proportional fitness selection* is performed. A distinguishable feature of the GP algorithm is that, for adding diversity to the population, is performed only one operator between crossover and mutation. The most common recombination operator (crossover) exchanges two randomly chosen sub-trees among the parents. In the case of mutation, it typically replaces a randomly chosen sub-tree with a randomly generated tree. In both cases, the size of the child can exceed the size of the parent and this can lead to a significant increase in the depth of the tree.

Algorithm 1 Genetic Programming

Input: population size n_{pop}, generation number n_{gen}, min and max depth for tree initialization d_{min} and d_{max}, tree generation function G, probabilities of crossover and mutation p_{cross} and p_{mut}, min and max depth for sub-tree mutation m_{min} and m_{max}, selection algorithm F_{sel}
Output: the best individual

```
P = INITIALIZE(n_pop, d_min, d_max, G)
for iteration = 1, 2, ..., n_gen do
    initialize S to empty set
    S = F_sel(P)
    for individual s in S do
        perform crossover(s) with probability p_cross
        perform mutation(s, m_min, m_max) with probability p_mut
        apply bloat control on s
    end for
    P ← S
end for
return best individual s ever lived
```

This problem is the so-called *bloat* (survival of the fattest). To overcome this problem, a penalty term can be included in the fitness function which discourages large trees and prohibits variation operators that would deliver too big children [2].

In this study, we adopt canonical GP according to what is written above. In addition, parameters and implementation details will be discussed in the following chapter.

4

Chapter 4: Experimental Results

In this chapter, we report the details of the experiment and the final performance achieved. It is provided a description of the dataset on which the experiment has been performed. Subsequently, the design of the planned workflow for the experiment is presented. Finally, more details about each phase of the work are provided in the next sections.

4.1. Dataset Analysis

The DARWIN project dataset has been initially used to evaluate the performance of the developed work. It includes 25 different tasks, which are grouped into four categories that are: *Graphic tasks*, *Copy and Reverse Copy tasks*, *Memory tasks*, *Dictation*. The dataset has been collected thanks to 174 individuals, of which 89 are Alzheimer's patients, and 85 are healthy subjects. For more details on the dataset check [9].

Thus, each sample is represented by a feature vector of 450 real values in CD. On the contrary, the RD includes 107 features for each sample and it was obtained by applying the feature selection method. Features are crucial in creating predictive models. However, having too many features may lead to overfitting and also to a multitude of problems related to the curse of dimensionality. For this reason, it has been applied the *Feature Importance* technique on CD to figure out which are the most important features to use in the classification phase. In [9], starting from CD, the authors extract and remove highly-correlated features, using the *Searching for Uncorrelated List of Variables* (SULOV). To get more in-depth knowledge about the SULOV selection, see [9].

4.2. Experimental Design

In this study, the objective of the experiment is to measure the performance of the GP Algorithm in terms of accuracy, specificity, and sensitivity. In addition, thanks to the selection property of the GP Algorithm, we want to figure out what features are more influential and what's their role in the decision process of classification and they're impact on the performance. Furthermore, thanks to the interpretability property of GP, we want to study the generated solutions and analyze how the classification works. To achieve the aims just mentioned, the experiment has been divided into the following phases:

1. A tuning phase has been performed on CD and RD to find the best values for the hyperparameters[1] and the best function set. A bloat control method has been used to avoid the overgrowth of the trees. More details on the implementation and performance evaluation method will be provided respectively in Section 4.3 and Section 4.6;

[1] The definition of the hyperparameters can be found in the input of Algorithm 1

2. Once the best configuration of hyperparameters has been tested, we want to analyze the number of occurrences of each feature to perform a further feature reduction out of the best performing dataset, as described in Section 4.5.
3. Then, we want to evaluate the performance of the same hyperparameters configuration on the new dataset, built after the most relevant features selected in the previous step. Then, we compare the results with the state-of-the-art and analyze the resulted solutions.

Figure 1 illustrates the workflow planned for the execution of the experiment.

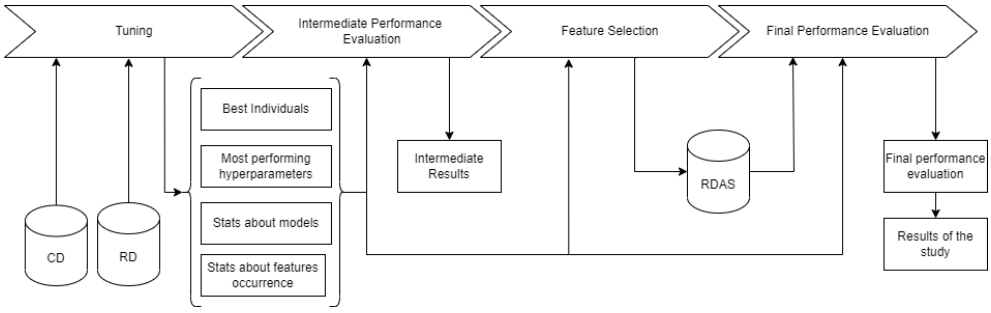


Figure 1: Designed workflow for the experiment

4.3. Implementation Details

The algorithm for GP has been implemented in Python. Each step of the GP breeding has been performed through the functions of the library *DEAP*. The generation of the random seeds used during the process has been performed through the function of the built-in library *random*. The operators used for the function set are retrieved from the library *operator*. Nonparametric tests have been conducted with the library *scipy*. Graphics and visual representations have been produced through the libraries *networkx*, *matplotlib* and *seaborn*.

4.4. Tuning Phase Results

Firstly, a tuning phase was performed in which different hyperparameter values were tested to generate individuals with the best fitness (accuracy on the Training Set). The best values are reported in Table 1. Considering the dataset provides numeric features derived from the analysis of handwriting tasks, the main operators tested for the function set are mostly mathematical and comparative operators. Table 2 reports the best operators according to the performance retrieved in this phase. As detailed in Table 3, GP performed better on RD than CD as [9] proved to be also true for other methods such as SVM, RF, and RNSA.

4.5. Feature Selection

Attempting to achieve better performance, we apply a further feature reduction, by using the selection property of GP. The method works as it follows: analyzing the number of times each feature occurs in the best trees, we picked the top 10 occurring features. The Table 4 resumes those most relevant features, ordered by the average number of occurrences per tree[2], showing first the most recurrent feature (a detailed description of the features and the tasks is available respectively in [9] and [3]). Starting from them, we built the *Reduced Dataset After Selection* (RDAS). In addition, it's known that the

[2] $OccurrenceRate = \frac{NumberOfOccurrences}{NumberOfTrees}$

Table 1: Chosen Hyperparameters

Parameter	Value
n_{pop}	1200
n_{gen}	40
d_{min}	0
d_{max}	1
G	Full generation
p_{cross}	0.5
p_{mut}	0.2
m_{min}	0
m_{max}	1
F_{sel}	Tournament Selection with $t_{size} = 3$

Table 2: Function Set

Function	Definition	Arity
Addition	$OUT = a_1 + a_2$	2
Subtraction	$OUT = a_1 - a_2$	2
Greater than	**if** $a_1 > a_2$: $OUT = True$ **else** $OUT = False$	2
If than else	**if** a_1: $OUT = a_2$ **else** $OUT = a_3$	3

Table 3: Performance comparison between Complete Dataset (CD) and Reduced Dataset (RD)

	CD	**RD**
Average Accuracy	75.75%	78.77%
Accuracy Standard Deviation	7.41%	6.98%
Max Accuracy	91.37%	96.55%
Min Accuracy	56.89%	48.27%
Average Sensitivity	75.42%	79.49%
Average Specificity	76.35%	78.24%

Table 4: Details of Features Selected (in decreasing order of importance)

Name	Feature	Task	Occurrence Rate
x_0	Max X Extension	22	2.52
x_1	Air Time	17	1.37
x_2	Total Time	24	0.89
x_3	Total Time	9	0.85
x_4	Air Time	23	0.69
x_5	Air Time	6	0.62
x_6	Max Y Extension	21	0.55
x_7	Air Time	4	0.55
x_8	Total Time	16	0.42
x_9	Max Y Extension	2	0.32

features in CD are divided into six classes which are: temporal, kinematic, pen-down, spatial, pressure, and tremor measurement. The performed feature selection shows that temporal and spatial features tend to appear more in the best-generated trees. It is interesting to note that this consideration can help to build simpler diagnostic tests that take into account fewer factors.

4.6. Performance Evaluation

As regards the evaluation of the performance of the GP Algorithm, the chosen method was a *K-fold cross-validation* considering 6 folds. For each fold combination, 100 initialization seeds have been tested. The whole process has been repeated 6 times, one for each different way of splitting the dataset and generating the 100 seeds. The final results have been calculated on all the 3600 runs executed. Table 5 reports the final results obtained by the described process both on RD and RDAS. It is necessary to state that the results related to RF, SVM, and RNSA have been retrieved in a different context with a different approach. Refer to [9] for more information.

The results show that mean accuracy and the standard deviation of GP are comparable with state of the art. According to the results, GP performed better on RDAS than RD, proving that the further feature reduction increased the performance. Furthermore, a statistical analysis based on the Mann-Whitney nonparametric test [7] has been executed to demonstrate the improvement on RDAS with respect to RD, despite it being small, is an effective consequence of the feature reduction. Results are gathered on two distributions based on 3600 samples of accuracy each. The resulted p-value $p = 5.25 \times 10^{-9}$ proved that we can reject the H_0 hypothesis, and so the distributions are not the same, with enough confidence ($p < 0.05$). $U_1 = 6995892$ and $U2 = 5964108$ are the U values calculated from the test. Figure 2 shows a plot of the distributions of the models.

More detailed statistics obtained during the performance evaluation on the training set (TRN) and test set (TST) are available in Table 6.

According to the comparison with state-of-the-art solutions, GP is less performing than the confronted methods, however, the loss in terms of accuracy can be compensated by the *interpretability* property of the models generated. The result includes trees with an average depth of 7.5 which leads to simple and readable solutions. Analyzing the trees obtained from this approach, it is possible to get some hints on the reason why a certain subject has been classified in a certain way. Two of the best rules produced by the GP on RDAS are formalized as Algorithm 2 and Algorithm 3.

Table 5: Final Results

	RF	SVM	RNSA	GP	
	RD	RD	RD	RD	RDAS
Average Accuracy	91.62%	87.62%	96.58%	78.77%	80%
Accuracy Standard Deviation	5.15%	4.66%	1.82%	6.98%	7.05%
Max Accuracy	100%	94.29%	98.32%	96.55%	100%
Min Accuracy	82.86%	77.14%	92.44%	48.27%	51.72%
Average Sensitivity	88.88%	88.88%	100%	79.49%	78.45%
Average Specificity	94.11%	88.23%	86.67%	78.24%	81.52%

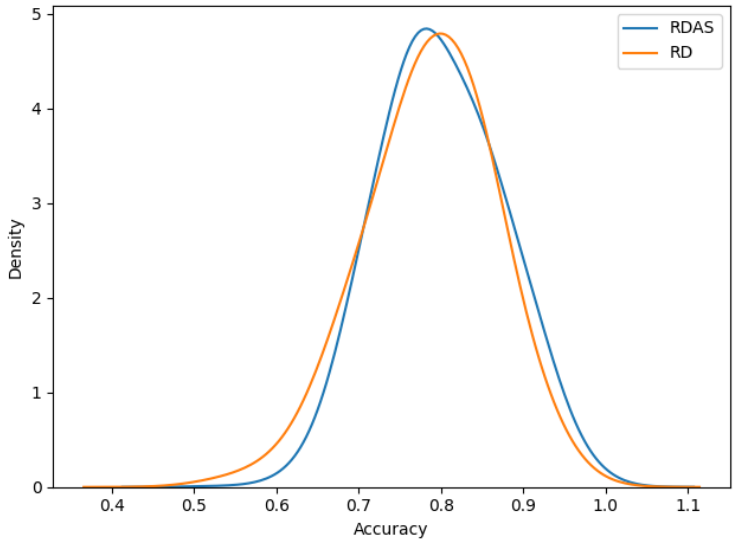


Figure 2: Distribution on RD and RDAS

Table 6: Performance of GP on Accuracy, Sensitivity and Specificity got from Training Set (TRN) and Test Set (TST)

	RD		RDAS	
	TRN	TST	TRN	TST
Average Accuracy	86.52%	78.77%	89.63%	80%
Accuracy Standard Deviation	2.50%	6.98%	1.76%	7.05%
Max Accuracy	94.48%	96.55%	95.17%	100%
Min Accuracy	75.17%	48.27%	84.82%	51.72%
Average Sensitivity	86.35%	79.49%	88.15%	78.45%
Sensitivity Standard Deviation	6.08%	11.91%	4.85%	11.82%
Max Sensitivity	98.70%	100%	100%	100%
Min Sensitivity	59.15%	26.66%	74.72%	35.71%
Average Specificity	86.65%	78.24%	91%	81.52%
Specificity Standard Deviation	5.63%	11%	3.70%	9.19%
Max Specificity	100%	100%	100%	100%
Min Specificity	63.38%	26.66%	75.71%	41.66%

Algorithm 2 Tree #1

Input: x_0, x_2, x_4, x_5, x_6, x_8, x_9
Output: The diagnosis

if $x_4 > x_8$ **then** $y_3 = x_0$ **else** $y_3 = x_4$
if $y_3 > x_8$ **then** $y_2 = x_0 - x_9$ **else** $y_2 = x_6 - x_9$
if $x_0 > x_6$ **then** $y_1 = y_2$ **else** $y_1 = x_4$
if $x_8 > y1 + x_5 - x_9$ **then**
    $OUT =$ "Healthy"
else
    $OUT =$ "Ill"
end if

Algorithm 3 Tree #2

Input: x_0, x_1, x_2, x_3, x_4, x_5, x_7, x_8
Output: The diagnosis

if $x_0 > x_8$ **then** $y_3 = x_4$ **else** $y_3 = x_3$
if $x_5 > x_8$ **then** $y_2 = x_3$ **else** $y_2 = x_4 - x_7$
if $x_0 > x_8$ **then** $y_1 = x_0$ **else** $y_1 = y_2 - y_3$
if $4x_8 + x_1 - x_0 - x_2 - x_3 - y_1 > x_5$ **then**
    $OUT =$ "Healthy"
else
    $OUT =$ "Ill"
end if

5

Conclusions

The motivation for the study presented in this paper was to find a new method to generate interpretable solutions that can be applied especially in the medical field for supporting the diagnosis of Alzheimer's Disease. Despite Genetic Programming being less performing than other techniques explored in state of the art, it proved to be a powerful way to get new automatic tools that can support medical staff in their decision-making process. It can also give hints to discover correlations between symptoms and disease, even though it works on a kind of data that is far easier to collect with respect to traditional methods of acquiring data typically used for an Artificial Neural Network.

Given the way data are collected, this may lead to the development of some user-friendly tools for the continuous monitoring of aging individuals such as tablet applications.

Even if the results are promising, it must be taken into account the experimental setting. A possible extension of this work could include the improvement of the robustness of the method and the affidability of the test. Another future development could be to verify the potential of the method on a wider dataset and with new different function sets. In addition, it could also be worth to try merging Genetic Programming with other techniques that proved to be valid such as the Negative Selection Algorithm.

References

[1] Irene Azzali et al. “Vectorial GP for Alzheimer’s Disease Prediction Through Handwriting Analysis”. In: *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*. Springer. 2022, pp. 517–530.

[2] Anthony Brabazon, Michael O’Neill, and Seán McGarraghy. *Natural computing algorithms*. Vol. 554. Springer, 2015, pp. 95–105.

[3] Nicole Dalia Cilia et al. “An experimental protocol to support cognitive impairment diagnosis by using handwriting analysis”. In: *Procedia Computer Science* 141 (2018), pp. 466–471.

[4] Nicole Dalia Cilia et al. “Handwriting analysis to support alzheimer’s disease diagnosis: a preliminary study”. In: *International Conference on Computer Analysis of Images and Patterns*. Springer. 2019, pp. 143–151.

[5] Nicole Dalia Cilia et al. “Using handwriting features to characterize cognitive impairment”. In: *International Conference on Image Analysis and Processing*. Springer. 2019, pp. 683–693.

[6] Leandro Nunes De Castro. *Fundamentals of natural computing: basic concepts, algorithms, and applications*. Chapman and Hall/CRC, 2006, pp. 105–112.

[7] Yadolah Dodge. *The concise encyclopedia of statistics*. Springer Science & Business Media, 2008.

[8] Josep Garre-Olmo et al. “Kinematic and pressure features of handwriting and drawing: preliminary results between patients with mild cognitive impairment, Alzheimer disease and healthy controls”. In: *Current Alzheimer research* 14.9 (2017), pp. 960–968.

[9] De Gregorio Giuseppe, Della Cioppa Antonio, and Marcelli Angelo. “Negative Selection Algorithm for Alzheimer’s Diagnosis: Design and Performance Evaluation”. In: (2022), pp. 531–546.

[10] Tatsuya Ishikawa et al. “Handwriting Features of Multiple Drawing Tests for Early Detection of Alzheimer’s Disease: A Preliminary Result.” In: *MedInfo*. 2019, pp. 168–172.

[11] Christian Kahindo et al. “Characterizing early-stage Alzheimer through spatiotemporal dynamics of handwriting”. In: *IEEE Signal Processing Letters* 25.8 (2018), pp. 1136–1140.

[12] Ananya Mandal. *What is a Biomarker?* 2019. URL: https://www.news-medical.net/health/What-is-a-Biomarker.aspx (visited on 02/26/2019).

[13] Mahda Nasrolahzadeh, Shahryar Rahnamayan, and Javad Haddadnia. “Alzheimer’s disease diagnosis using genetic programming based on higher order spectra features”. In: *Machine Learning with Applications* 7 (2022), p. 100225.

[14] Antonio Parziale et al. “A decision tree for automatic diagnosis of parkinson’s disease from offline drawing samples: experiments and findings”. In: *International Conference on Image Analysis and Processing*. Springer. 2019, pp. 196–206.

[15] Antonio Parziale et al. “Cartesian genetic programming for diagnosis of Parkinson disease through handwriting analysis: Performance vs. interpretability issues”. In: *Artificial Intelligence in Medicine* 111 (2021), p. 101984.

[16] Giuseppe Pirlo et al. “Early diagnosis of neurodegenerative diseases by handwritten signature analysis”. In: *International Conference on Image Analysis and Processing*. Springer. 2015, pp. 290–297.

[17] Martin James Prince et al. “World Alzheimer Report 2015-The Global Impact of Dementia: An analysis of prevalence, incidence, cost and trends”. In: (2015).

[18] Perla Werner et al. “Handwriting process variables discriminating mild Alzheimer’s disease and mild cognitive impairment”. In: *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences* 61.4 (2006), P228–P236.