EasyChair Preprint

№ 3785

Parkinson's Disease Diagnosis: Towards Grammar-based Explainable Artificial Intelligence

Federica Cavaliere, Antonio Della Cioppa, Angelo Marcelli, Antonio Parziale and Rosa Senatore

July 7, 2020

Parkinson's Disease Diagnosis: Towards Grammar-based Explainable Artificial Intelligence

F. Cavaliere, A. Della Cioppa, A. Marcelli, A. Parziale, R. Senatore
NCLab, DIEM, University of Salerno
Salerno, ITALY

***Abstract*—The basic technology that reinvents machines to personalize human experiences is Machine Learning (ML), a branch of Artificial Intelligence (AI) and a strong buzzword in today's digital world. Despite its success, the most significant limitation of ML is the lack of transparency behind its behavior, which leaves users with a poor understanding of how it makes decisions, such it is the case for Deep Learning models. If the final user does not trust a model, he will not use it. This is especially true in medical diagnosis practice: physicians cannot simply use the predictions of the model but must trust the results it provides. This work focuses on the automatic early detection of Parkinson's disease (PD), whose impact on both the individual's quality of life and social well-being is constantly increasing with the aging of the population. To this end, we propose an explainable approach based on Genetic Programming, called Grammar Evolution (GE). This technique uses context-free grammar to describe the language of the programs to be generated and evolved. In this case, the generated programs are the explicit classification rules for the diagnosis of the subjects. The results of the experiments obtained on the publicly available HandPD data set show GE's high expressive power and performance comparable to those of several ML models that have been proposed in the literature.**

***Index Terms*—Explainable Artificial Intelligence, Grammatical Evolution, Parkinson's Disease, Supervised Learning by Classification, e-Health.**

I. Introduction

Machine Learning (ML), a branch of Artificial Intelligence (AI), has gained increasing interest from the scientific community, since it allows to support humans in the decision-making processes regarding several fields. However, despite their success, ML based approaches have some limitations and disadvantages [1]. Among them, the poor explainability of ML complex models has hampered their extensive application in those fields where the impact of the decision is critical, as it is the case for medical application. Indeed, the more a decision can influence people's lives, the more important it is to understand which are the factors that lead to that particular decision. Physicians cannot simply use the predictions of the model but should also trust the results obtained. To trust them, physicians have to understand how and why these decisions were made and compare these reasons with their previous domain knowledge. Consequently, the black-box approach implemented by most of ML classification systems, which do not clearly indicate the adopted decision rules, has hampered their use by clinicians.

In order to tackle this issue, here we adopted an evolutionary technique based on Genetic Programming (GP), called Grammatical Evolution (GE)[2], which allows to derive automatically the explicit rules exploited for pattern classification in a decision tree form, that makes them highly interpretable.

We applied GE for the automatic diagnosis of Parkinson's disease(PD) through handwriting analysis. In particular, we evaluated the GE approach on a publicly available dataset, the HandPD [3], which contains handwriting samples drawn by healthy subjects and PD patients. Therefore, the aim of this work is two-fold: evaluate whether the GE approach is able to ensure good classification performance while providing explicit decision rules, and provide some insights for the design of a non-invasive, inexpensive and quick-to-administer diagnostic tool for the diagnosis of PD.

PD affects millions of people around the world, and one of its main symptoms is the impairment of the motor abilities. Mostly used diagnostic tools include psychometric tests, neuroimaging, and cerebrospinal fluid examination, though these methods are time-consuming, expensive and invasive, respectively.

Handwriting production, being a motor task that involves skilled control of complex movements, is influenced by the activity of the basal ganglia [4], which is compromised in people affected by the disease [5]. The analysis of handwriting in people with PD has led to a greater understanding of the motor processes occurring in pathological and physiological conditions [6, 7, 8, 9, 10].

Furthermore, the use of handwriting analysis for supporting the diagnosis or assessing the stage of the disease is receiving increasing interest in the last decades[11]. Several ML techniques have been exploited for the automatic classification of handwriting samples contained in the HandPD dataset. Most of previous work have applied standard ML classification approaches (such as Gaussian Naive Bayes, K-Nearest Neighborhood, Support Vector Machine, Decision Tree, and AdaBoost Classifier) [3, 12], exploited Deep or Convolutional Neural Network[13, 14]. and combined feature selection algorithms to the classification methods [15, 16]. However,

the classification techniques adopted in previous work are characterized by reduced level of interpretability of the exploited classification rules. In previous work we have already afforded the explainability problem related to the standard ML approaches, and investigated the performance of another GP approach, called Cartesian Genetic Programming (CGP), which allows to both classify handwriting samples and obtain explicitly the classification criteria adopted [17, 18].

With the same aim, here we have investigated the performance of the GE approach, combined with the use of Automatically Defined Functions (ADFs) [19], trying to manage the trade-off between performance and explainability once the design is guided by the latter.

The remainder of the work is organized as follows. Section II-A describes the GE technique, discusses the ADF's concept and reports the design of the grammars chosen to solve the problem under consideration. Section III describes the parameter configuration of GE and the data set used in the experiments, reports the achieved performance, and discuss obtained results. Eventually, section IV draws the conclusions and sketches some future steps of this work.

II. Method

A. Grammatical Evolution

Grammatical Evolution is a population-based evolutionary algorithm that allows the automatic generation of programs by exploiting the principles from molecular biology to the representational power of formal grammars [2]. It is a grammar-based form of Genetic Programming [20] that adopts a genotype-to-phenotype mapping process in order to generate the output programs.

GE does not perform the evolutionary process on the actual programs, which are explicitly encoded by individuals as in the case of Genetic Programming, but rather on variable-length binary strings representing the individuals of population. The construction of a syntactically correct program from a binary string is guided by the aim of maximizing a fitness function.

The mapping process requires the definition of a Backus Naur Form (BNF) grammar in order to evolve code in any language, and ensures its syntactic correctness.It allows to express grammar of a language in the form of production rules [21]. In BNF a grammar is represented by the tuple $\{N, T, P, S\}$, where N is the set of *nonterminals*, T is the set of *terminals*, P is the set of production rules that maps the elements of N to T and S is the starting nonterminal symbol. The selection of an appropriate production rule is executed starting from the 8-bits codons of the genome, which are represented by integer values. In particular, a rule is selected by using the eq. (1):

$$R = |R_{NT}|\%C_i \tag{1}$$

where R is the chosen production rule, $|R_{NT}|$ is the number of rules for current non-terminal, % is the symbol of modulo operation, C_i is the codon non-negative integer value, i.e. an element of chromosome array C.

Each time a production rule has to be selected to map from a nonterminal, another codon is read. In this way, the system traverses the genome and the starting chromosome (array) is transformed into a grammatically correct function. During the genotype-phenotype mapping process it is possible that individuals run out of codons and in this case, individual is wrapped and codons are reused. This processing draws inspiration from the phenomenon of gene overlap, which has been observed in many organisms. In GE, every time the same codon is expressed, it will always generate the same integer value, but depending on the current non-terminal to which it is applied, it can involve the selection of a different production rule. Nevertheless, the process guarantees that every time the genotype of an individual is mapped to its phenotype, the same output is generated.

The automatic generated program that perform better in term of a fitness function has a greater chance of transmitting its DNA to the next generation of chromosomes. From this point of view, GE does not differ much from other evolutionary algorithms, since the chromosomes are crossed and mutated to generate the next generation of chromosomes, which are then mapped on the grammar and tested with respect to the fitness function.

B. Automatically Defined Functions

A common strategy adopted for solving complex problems is the *divide-et-impera* approach: the original problem is splitted in smaller and more easy to solve tasks whose solutions are combined hierarchically to solve the problem as a whole [22].

Automatically Defined Functions are the tool adopted in GP to allow the decomposition of large problems into modules [19]. It has been shown that GP equipped with ADFs outperforms standard GP on many problems [23].

The adoption of ADFs for GE allows the evolution of both the functions and the main program that calls the evolving functions. ADFs allow a dynamic grammatical approach: multiple functions are defined and they are created dynamically with a variable number of arguments. A grammar can be further deepened by allowing ADFs to recursively call themselves or other ADFs in order to increase their expressiveness.

C. Classification Rule

The aim of this paper is to define a classifier for the diagnosis of Parkinson's disease that is easily explainable and highly performing. A general, simple and performing classification rule produces a binary diagnosis f as linear combination of simpler binary classifiers adf, each of which evaluates a single feature x. Equation 2 synthesizes the classification rule f weighting the effect of each feature:

$$f = a_1 \cdot adf_1(x_1) + a_2 \cdot adf_2(x_2) + ... + a_N \cdot adf_N(x_N) > T \tag{2}$$

where f is the overall binary diagnosis (if true, the subject is classified as parkinsonian, otherwise as healthy), N is the number of features, $x_1, ..., x_N$ are the features, T is a positive threshold, $a_1, ..., a_N$ are weights of each feature's function and $adf_1, ..., adf_N$ are binary diagnoses taken on each feature.

D. Grammar design

Each individual of the population is a decision rule that has explicitly defined the weights, the functions and the threshold. It follows that, as the population evolves from generation to generation, the internal structure of the functions, the weights and the threshold value are optimized during the evolution with the aim of maximizing the fitness function. In particular, we chose as fitness function the accuracy of the classifier.

The way individuals evolve depends on how the grammar was designed and a great attention was paid to the complexity of the internal structure of the ADFs. We have defined two grammars that are able to produce instances of the general classification rule described by eq. 2) whose difference is in the design of the ADFs. For both grammars, ADFs have a tree structure similar to a decision tree so that they could be easily understood by humans. In particular, each function has a tree structure consisting of a recursive chain of *if-else* constructs and return declarations. What really changes between the two grammars is, for each ADF, the maximum depth of each tree and the complexity of the conditions of the constructs inside the tree. The grammar based on deeper and complex ADF was named *Full* while the other one, based upon ADF with simpler constructs was named *Reduced*. Both the grammars share the set of production rules reported in Fig. 1 while Fig. 3 and Fig. 2 report the production rules for the *Reduced* and the *Full* grammar, respectively. The comparison of performance of both the grammars is helpful to understand if a tradeoff exists between performance and interpretability.

The *Reduced grammar* limits the nesting of *if-else* constructs to a maximum of 5 and, therefore, it defines the maximum depth reachable by each ADF. The constructs of such trees consist only of conditions of comparison between the pure values of the features (already non-negative numbers) and non-negative numbers.

The *Full grammar* doesn't limit the maximum depth that can be reached by a tree. In this way, evolution may bring out trees with a depth greater than five if they lead to more performing solutions. Furthermore, the constructs of such trees consist only of conditions of comparison between the values of the characteristics or functions of them and non-negative numbers.

III. Experimental setup and results

A. GE implementation

The GE was implemented using PonyGE2 [24]. Table I reports the value of the parameters used in the experiments. The parameters that influence the genotype-phenotype mapping process are the maximum value assumed by a genome element (the Codon size), and the maximum genome size (Max Genome Length). The choice of these parameters depends directly on the grammar chosen. The codon size depends on the maximum number of choices, i.e. the production activities. Since we set that the value of the maximum number of choices to 10, and considering that the maximum codon size

⟨*p*⟩ ::= ⟨*defp*⟩{::}⟨*callp*⟩
⟨*defp*⟩ ::= def p():{:{::} ⟨*defadf0*⟩ {::} ⟨*defadf1*⟩ {::} ⟨*defadf2*⟩ {::} ⟨*defadf3*⟩ {::} ⟨*defadf4*⟩ {::} ⟨*defadf5*⟩ {::} ⟨*defadf6*⟩ {::} ⟨*defadf7*⟩ {::} ⟨*defadf8*⟩ {::} ⟨*codep*⟩:}
⟨*callp*⟩ ::= result=p()
⟨*defadf0*⟩ ::= def adf0(index):{:{::}⟨*code*⟩{::}:}
⟨*defadf1*⟩ ::= def adf1(index):{:{::}⟨*code*⟩{::}:}
⟨*defadf2*⟩ ::= def adf2(index):{:{::}⟨*code*⟩{::}:}
⟨*defadf3*⟩ ::= def adf3(index):{:{::}⟨*code*⟩{::}:}
⟨*defadf4*⟩ ::= def adf4(index):{:{::}⟨*code*⟩{::}:}
⟨*defadf5*⟩ ::= def adf5(index):{:{::}⟨*code*⟩{::}:}
⟨*defadf6*⟩ ::= def adf6(index):{:{::}⟨*code*⟩{::}:}
⟨*defadf7*⟩ ::= def adf7(index):{:{::}⟨*code*⟩{::}:}
⟨*defadf8*⟩ ::= def adf8(index):{:{::}⟨*code*⟩{::}:}
⟨*f*⟩ ::= ⟨*sign*⟩⟨*a*⟩*adf0(0) | ⟨*sign*⟩⟨*a*⟩*adf1(1) | ⟨*sign*⟩⟨*a*⟩*adf2(2) | ⟨*sign*⟩⟨*a*⟩*adf3(3) | ⟨*sign*⟩⟨*a*⟩*adf4(4) | ⟨*sign*⟩⟨*a*⟩*adf5(5) | ⟨*sign*⟩⟨*a*⟩*adf6(6) | ⟨*sign*⟩⟨*a*⟩*adf7(7) | ⟨*sign*⟩⟨*a*⟩*adf8(8)
⟨*codep*⟩ ::= return ⟨*f*⟩ > ⟨*number*⟩
⟨*sign*⟩ ::= + | -
⟨*bool*⟩ ::= True | False
⟨*c*⟩ ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Fig. 1: Production rules belonging to both the Reduced and the Full Grammar.

⟨*code*⟩ ::= ⟨*if-else*⟩ | if ⟨*cond*⟩:{:return ⟨*bool*⟩:} {::}else:{:return ⟨*bool*⟩:}
⟨*if-else*⟩ ::= if ⟨*cond*⟩ :{:{::} ⟨*code*⟩ {::} return ⟨*bool*⟩ :}{::} else:{:{::}⟨*code*⟩{::} return ⟨*bool*⟩:}
⟨*cond*⟩ ::= ⟨*arg*⟩>⟨*sign*⟩⟨*number*⟩ | ⟨*arg*⟩<⟨*sign*⟩⟨*number*⟩ | ⟨*arg*⟩==⟨*sign*⟩⟨*number*⟩
⟨*arg*⟩ ::= ⟨*fun*⟩(x[index]) | x[index]
⟨*fun*⟩ ::= sin | cos | tanh | sigmoid | rlog | psqrt
⟨*number*⟩ ::= ⟨*c*⟩⟨*c*⟩⟨*c*⟩⟨*c*⟩⟨*c*⟩⟨*c*⟩⟨*c*⟩.⟨*c*⟩⟨*c*⟩⟨*c*⟩⟨*c*⟩
⟨*a*⟩ ::= ⟨*c*⟩⟨*c*⟩⟨*c*⟩.⟨*c*⟩⟨*c*⟩⟨*c*⟩

Fig. 2: Production rules belonging only to Full Grammar.

⟨*code*⟩ ::= ⟨*p0*⟩ | ⟨*p1*⟩ | ⟨*p2*⟩ | ⟨*p3*⟩ | ⟨*p4*⟩
⟨*p0*⟩ ::= if ⟨*cond*⟩ :{: return ⟨*bool*⟩ :}{::} else :{: return ⟨*bool*⟩ :}
⟨*p1*⟩ ::= if ⟨*cond*⟩ :{: ⟨*p0*⟩ :}{::} else :{: ⟨*p0*⟩:}
⟨*p2*⟩ ::= if ⟨*cond*⟩ :{: ⟨*p1*⟩ :}{::} else :{: ⟨*p1*⟩:}
⟨*p3*⟩ ::= if ⟨*cond*⟩ :{: ⟨*p2*⟩ :}{::} else :{: ⟨*p2*⟩:}
⟨*p4*⟩ ::= if ⟨*cond*⟩ :{: ⟨*p3*⟩ :}{::} else :{: ⟨*p3*⟩:}
⟨*cond*⟩ ::= x[index] > ⟨*number*⟩ | x[index] < ⟨*number*⟩ | x[index]==⟨*number*⟩
⟨*number*⟩ ::= ⟨*c*⟩⟨*c*⟩⟨*c*⟩⟨*c*⟩⟨*c*⟩.⟨*c*⟩⟨*c*⟩⟨*c*⟩⟨*c*⟩⟨*c*⟩
⟨*a*⟩ ::= ⟨*c*⟩⟨*c*⟩.⟨*c*⟩⟨*c*⟩

Fig. 3: Production rules belonging only to Reduced Grammar.

TABLE I: Parameters Value.

Parameter	Value
Population size	500
Codon size	100
Crossover probability	0.75
Generations (Reduced grammar)	500
Generations (Full grammar)	2000
Maximum Genome Length	500
Maximum Initial Tree Depth	10
Maximum Tree Depth	90
Selection	Tournament
Tournament size	2

should not be much larger, we choose 100 as the value for the maximum codon size. As regarding the maximum genome size, higher values could cause the generation of more complex rules (i.e. many nested *if-else* constructs in the ADFs), with the consequence that such rules could not emerge during the evolution. On the other hand, reduced values of the maximum genome size could limit the possibility of improving the fitness of the population during the evolution, since it could cause the generation of offspring made up of many invalid individuals (i.e. individuals with incomplete mapping). Consequently, we have evaluated a set of starting values, and different increase amounts, adapting to the evolutionary process (adaptive maximum genome size), and observed that using a starting value of 500 for the maximum genome size provides the best performance on the training test. Furthermore, best performance is achieved for a size increase of 15 every 50 generations of evolution stagnation, or 50 generations in which an individual shows better fitness compared to the last best.

B. Dataset

The experimentation has been performed using the HandPD dataset [25]. HandPD is a public dataset that consists of images extracted from handwriting samples of 92 individuals: 18 healthy people (Healthy Group) and 74 affected by Parkinson's disease (Patients Group).

TABLE II: Features description of the dataset used. HT: handwritten trace, ET: exam template.

Feature	Description
x_0	RMS of the difference between HT and ET radius: $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(r^i_{HT} - r^i_{ET})^2}$
x_1	Maximum difference between HT and ET radius: $\Delta_{max} = \text{argmax}_i(\lvert r^i_{HT} - r^i_{ET}\rvert)$
x_2	Minimum difference between HT and ET radius: $\Delta_{min} = \text{argmin}_i(\lvert r^i_{HT} - r^i_{ET}\rvert)$
x_3	Standard Deviation of the difference between HT and ET radius
x_4	Mean Relative Tremor: $\frac{1}{n-10}\sum_{i=1}^{n}(\lvert r^i_{HT} - r^{i-9}_{ET}\rvert)$
x_5	Maximum HT radius
x_6	Minimum HT radius
x_7	Standard Deviation of HT radius
x_8	Number of times the difference between HT and ET radius changes sign

Each individual was asked to draw four spirals and four meanders. In particular, each individual received a form with printed templates of spirals and meanders (ET) and had to trace over them with a pencil producing handwritten traces (HT). The dataset also contains the features extracted for each image. In particular, each spiral or meander image was represented by the 9 features described in Table II and computed by comparing the template image with the handwritten trace.

C. Protocol

The performance of the two proposed grammars were evaluated by adopting a stratified k-fold cross-validation. This method guarantees that the division in training and test sets was done by preserving the sample distribution in each class. In particular, we evaluated the performance by choosing different values of k: 3, 7 and 10.

The stratified k-fold cross-validation was repeated 5 times by initializing the GE with 5 different seeds.

The GE was interrupted before the maximum number of generation was reached if one of the following events occurred: the accuracy of the best individual over the training set was greater or equal than 90%, or a stagnation of the evolution for more than 250 generations in case of Reduced grammar and more than 1000 in case of the Full one. In particular, the first condition is introduced for avoiding the overfitting.

D. Results

Table III reports the results obtained by evolving the Reduced and the Full grammars following the protocol described above. The performance are computed by evaluating spirals and meanders separately and by averaging over the 5 repetitions of the stratified k-fold cross-validation.

Table IV reports the ML approaches investigated in other work for automatically classify the handwriting samples of the HandPD dataset. For each work, the set of applied ML approaches and the best performing one is reported, with the corresponding mean accuracy obtained in classifying spirals and meanders samples and the number of features provided to the classifier. In [15] a chi-squared procedure has been applied for feature

TABLE III: Results obtained with both the grammars as the number k of folders varies.

	Reduced		Full	
Number of folders (k)	Spirals	Meanders	Spirals	Meanders
3	78.74 ± 0.26	79.10 ± 0.61	78.00 $\pm$4.96	75.77 ± 0.00
7	78.85 ± 0.28	79.78 ± 0.50	77.99 $\pm$4.96	75.77 ± 0.00
10	78.94 ± 0.44	79.77 ± 0.40	75.77 $\pm$0.00	75.77 ± 0.00

selection on the set of features defined in [3]. In [13] 2048 features have been extracted using ResNet-50 and used for classification. In [18] feature reduction has been automatically performed by the CGP approach, and thus 6 out of 9 features were included in the classification rules explicitly provided by the CGP. With the exception of the approach exploited in [18], other work used ML approaches that do not allow to explicitly extract the rules employed for obtaining the classification. Indeed, one of the key features of evaluating the performance of GE is the possibility of obtaining also some insights into the rationale behind the classification process, that could be used for providing efficient guidelines for the diagnosis of the disease. Furthermore, in terms of classification performance, the GE approach outperforms most of previous work, with the exception of [13]. It is noteworthy that the number of features exploited in this approach is much higher and makes the classification criteria hard to interpret, and the method is characterized by a low level of explainability.

IV. Conclusions

We have presented an approach to discriminate between healthy subjects and PD patients by analysing their handwriting production. The design was aimed at developing a method capable of providing classification rules that could be easily interpreted by the clinicians, as it is intended to help them in the early diagnosis of this type of neurodegenerative diseases. To achieve this aim, we have adopted GE for building the classifiers, as it provides classification rules in terms of chains of if-then rules, thus mimicking very closely the diagnostic process commonly adopted by clinicians. As it is widely recognized that there exists a trade-off between explainability and performance, we have designed a set of experiments for comparing the performance of our approach with the current state-of-the art on the HandPD dataset, which was adopted in previous work. The results reported in the previous section show that, when the same set of features is adopted, our approach exhibits better performance than its competitors, whose explainability is much lower. Only one of the method proposed in the literature performs better, but it works with a much larger set of features, and its explainability is much lower in comparison to GE. The experimental results also show that, differently from the other methods proposed in the literature, our approach does not exhibit significant difference in performance depending on the type of handwriting (either spirals or meanders) that is considered. By looking at the best decision rule provided by our approach during the k-fold cross validation (that are not reported here for space limitation) we observed that the most discriminant features for meanders are x_2 and x_0 that together are capable of correctly discriminating more than 84% of the samples. Moreover, the feature x_0 is the most discriminant feature for spiral and it correctly discriminates more than 78% of samples. The ability to encode the direction of movements aimed at reaching a sequence of targets involves the correct functioning of the basal ganglia, which is known to be impaired in people affected by PD. Here we found that, both in drawing spirals and meanders, the features that encode this ability are more discriminant than the others. Consequently, these results suggest that tests based on the analysis of handwriting production should include tasks that are specifically designed to test this ability, while tasks designed to highlight other features, such as the ability to keep the line of writing roughly horizontal or the presence of micrographia, may not be as effective as the former in discriminating PD from other neurodegenerative diseases or other conditions, such as stress and depression that affect handwriting production. As with regards to the two implementations of our approach, the experimental results suggest that the Reduced grammar achieves slightly better performance than the Full one on both spirals and meanders. As the former limits the depth of the tree, and therefore the length of the chain of rules leading to the diagnosis, this is a remarkable results, as it suggests that the most explainable classifier is also the top performing one, thus contradicting the general assumption that better performance requires more complex, and therefore less explainable, decision rules. Our current investigation is therefore focused on this issue to verify whether there are other paradigms based on evolutionary computation that exhibits this enjoyable characteristics.

References

[1] D. Mengnan, L. Ninghao, and Xia H. "Techniques for Interpretable Machine Learning". In: *Communications of the ACM* 63 (1 2020), pp. 68–77.

[2] M. O'Neill and C. Ryan. "Grammatical evolution". In: *IEEE Transactions on Evolutionary Computation* 4.5 (2001), pp. 349–358.

TABLE IV: Performance results obtained with other ML approaches.

		Spirals			Meanders		
Work	ML Approaches	Best Classifier	# of features	Mean Accuracy (%)	Best Classifier	# of features	Mean Accuracy (%)
[15]	Gaussian Naïve Bayes, DT KNN, SVM, AdaBoost	AdaBoost	2	72.46	AdaBoost	8	76.44
[13]	ResNet-50, SVM Bayes, OPF	OPF	2048	96.71	OPF	2048	96.31
[3]	Naïve Bayes, SVM, OPF	Naïve Bayes	9	64.23	SVM	9	66.72
[18]	CGP	CGP	9	57.51	CGP	9	76.60

[3] C. R. Pereira et al. "A new computer vision-based approach to aid the diagnosis of Parkinson's disease". In: *Computer Methods and Programs in Biomedicine* 136 (2016), pp. 79–88.

[4] R. Senatore and A. Marcelli. "A neural scheme for procedural motor learning of handwriting". In: *International Conference on Frontiers in Handwriting Recognition*. 2012, pp. 655–660.

[5] M.R. DeLong and T. Wichmann. "Circuits and Circuit Disorders of the Basal Ganglia". In: *Archives of Neurology* 64.1 (2007), pp. 20–24.

[6] J. E. McLennan et al. "Micrographia in Parkinson's disease". In: *Journal of the Neurological Sciences* 15.2 (1972), pp. 141–152.

[7] H.L. Teulings and G.E. Stelmach. "Control of stroke size, peak acceleration, and stroke duration in Parkinsonian handwriting". In: *Human Movement Science* 10.2-3 (1991), pp. 315–334.

[8] A. Van Gemmert et al. "Parkinson's disease and the control of size and speed in handwriting". In: *Neuropsychologia* 37 (1999), pp. 685–694.

[9] M. P. Broderick et al. "Hypometria and bradykinesia during drawing movements in individuals with Parkinson's disease". In: *Experimental Brain Research* 197.3 (2009), pp. 223–233.

[10] R. Senatore and A. Marcelli. "A paradigm for emulating the early learning stage of handwriting: Performance comparison between healthy controls and Parkinson's disease patients in drawing loop shapes". In: *Human Movement Science* 65 (2019), pp. 89–101.

[11] G. Vessio. "Dynamic Handwriting Analysis for Neurodegenerative Disease Assessment: A Literary Review". In: *Applied Sciences* 9.21 (Nov. 2019), p. 4666.

[12] A. 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.

[13] L. A. Passos et al. "Parkinson Disease Identification Using Residual Networks and Optimum-Path Forest". In: *IEEE 12th International Symposium on Applied Computational Intelligence and Informatics*. 2018, pp. 325–330.

[14] C. R. Pereira et al. "Deep Learning-Aided Parkinson's Disease Diagnosis from Handwritten Dynamics". In: *29th Conference on Graphics, Patterns and Images*. 2016, pp. 340–346.

[15] L. Ali et al. "Reliable Parkinson's Disease Detection by Analyzing Handwritten Drawings: Construction of an Unbiased Cascaded Learning System Based on Feature Selection and Adaptive Boosting Model". In: *IEEE Access* 7 (2019), pp. 116480–116489.

[16] P. Sharma et al. "Diagnosis of Parkinson's disease using modified grey wolf optimization". In: *Cognitive Systems Research* 54 (2019), pp. 100–115.

[17] R. Senatore, A. Della Cioppa, and A. Marcelli. "Automatic diagnosis of Parkinson disease through handwriting analysis: A cartesian genetic programming approach". In: *IEEE Symp. on Computer-Based Medical Syst.* 2019, pp. 312–317.

[18] R. Senatore, A. Della Cioppa, and A. Marcelli. "Automatic Diagnosis of Neurodegenerative Diseases: An evolutionary approach for facing the interpretability problem". In: *Information (Switzerland)* 10.30 (2019).

[19] E. Hemberg, M. O'Neill, and A. Brabazon. "An investigation into automatically defined function representations in Grammatical Evolution." In: *Mendel* (2009).

[20] J.R. Koza. *Genetic programming: on the programming of computers by means of natural selection*. Cambridge, MA, USA: MIT Press, 1992.

[21] Peter Naur et al. "Revised report on the algorithmic language Algol 60". In: *Communications of the ACM* 6.1 (1963), pp. 1–17.

[22] A. Newell, H. A. Simon, et al. *Human problem solving*. Prentice-Hall, NJ, 1972.

[23] John R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs*. Cambridge, MA, USA: MIT Press, 1994.

[24] M. Fenton et al. "PonyGE2: Grammatical Evolution in Python". In: *GECCO Conference*. ACM, 2017, pp. 1194–1201.

[25] C.R. Pereira et al. "A Step Towards the Automated Diagnosis of Parkinson's Disease: Analyzing Handwriting Movements". In: 28th IEEE Int. Symposium on Computer-Based Medical Systems. 2015, pp. 171–176.