

Constructing artificial features with Grammatical Evolution for the motor symptoms of Parkinson's Disease

Aimilios Psathas¹, Ioannis G. Tsoulos^{2,*}, Nikolaos Giannakeas³, Alexandros Tzallas⁴ and Vasileios Charilogis⁵

¹ Department of Informatics and Telecommunications, University of Ioannina, Greece; pint00141@uoi.gr

² Department of Informatics and Telecommunications, University of Ioannina, Greece; itsoulos@uoi.gr

³ Department of Informatics and Telecommunications, University of Ioannina, Greece; giannakeas@uoi.gr

⁴ Department of Informatics and Telecommunications, University of Ioannina, Greece; tzallas@uoi.gr

⁵ Department of Informatics and Telecommunications, University of Ioannina, Greece; v.charilog@uoi.gr

* Correspondence: itsoulos@uoi.gr

Abstract: This study introduces a set of features designed to capture the motor symptoms of Parkinson's Disease (PD) through detailed motion analysis, with a specific focus on how these symptoms change before and after medication. The features reflect key aspects of patient movement, such as tremor intensity, slowness, rigidity, instability, and irregularity in motion patterns. By quantifying how consistently and smoothly a patient performs specific tasks, the features provide insight into motor control quality and neurological function. Crucially, they are labeled according to the patient's state—before and after receiving medication—allowing for a clear comparison of treatment effects. This enables not only objective tracking of symptom severity, but also evaluation of medication responsiveness. These features address a fundamental clinical need: moving beyond subjective observation toward continuous, data-driven monitoring of disease progression and therapeutic effectiveness in Parkinson's Disease. In the current work the impact of feature construction using Grammatical Evolution on previously mentioned features is evaluated. We compare traditional neural architectures (MLP with ADAM, MLP with Genetic Algorithm, and RBF networks) against models trained on artificially constructed features. The results demonstrate a substantial reduction in classification error when 2 to 5 constructed features are used, achieving the lowest error rate (14.33%) with four generated features (FC4GEN), compared to 38.65% for the best baseline model (RBF). These findings highlight the effectiveness of evolutionary feature construction in enhancing classification accuracy.

Keywords: Machine learning; Evolutionary algorithms; Genetic Programming; Grammatical Evolution

Received:

Revised:

Accepted:

Published:

Citation: Psathas, A.; Tsoulos, I.G.;

Tzallas, A.; Charilogis, V.

Constructing artificial features with Grammatical Evolution for the motor symptoms of Parkinson's Disease.

Journal Not Specified **2025**, *1*, 0.

<https://doi.org/>

Copyright: © 2025 by the authors.

Submitted to *Journal Not Specified* for possible open access publication under the terms and conditions of the

Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

2. Materials and Methods

2.1. Preliminaries

2.2.

Grammatical Evolution procedure can be considered as a genetic algorithm, where the chromosomes are sets of randomly chosen integers. These integers represent production rules of the provided BNF grammar [1]. BNF grammars are usually expressed as sets having the form $G = (N, T, S, P)$, where

- The set N contains the non-terminal symbols of the grammar.
- The set T includes the terminal symbols of the grammar.
- S denotes the start symbol of the grammar, with $S \in N$.

Figure 1. An example of an extended BNF grammar.

```

S ::= <expr>          (0)
<expr> ::= (<expr> <op> <expr>) (0)
          | <func> ( <expr> )   (1)
          | <terminal>         (2)
<op> ::= +          (0)
        | -          (1)
        | *          (2)
        | /          (3)
<func> ::= sin      (0)
        | cos      (1)
        | exp      (2)
        | log      (3)
<terminal> ::= <xlist>          (0)
            | <dlist>.<dlist> (1)
<xlist> ::= x1      (0)
          | x2      (1)
          | .....
          | xd (d-1)
<dlist> ::= <digit>          (0)
          | <digit><digit>    (1)
          | <digit><digit><digit> (2)
<digit> ::= 0 (0)
          | 1 (1)
          | 2 (2)
          | 3 (3)
          | 4 (4)
          | 5 (5)
          | 6 (6)
          | 7 (7)
          | 8 (8)
          | 9 (9)

```

- P is the set of production rules of the grammar.

The Grammatical Evolution procedure utilizes an extended version of the initial BNF grammar by adding enumeration in the the production rules. As an example of an extended BNF grammar consider the grammar shown in Figure 1. The notation $\langle \rangle$ is used to enclose the non - terminal symbols of the grammar. The value d denotes the dimension of the input data. The Grammatical Evolution procedure starts from the start symbol of the program and gradually it creates valid programs in the underlying language, using the production rules of the grammar. The general scheme of the production procedure has as follows:

- **Get** the next element V from the chromosome that is under processing.
- **Select** the next rule using the equation $\text{Rule} = V \bmod \text{NR}$. The symbol NR denotes the number of production rules for the under processing non – terminal symbol.

A full working example of the production procedure is the chromosome

$$x = [9, 8, 6, 4, 16, 10, 17, 23, 8, 14]$$

with $d = 3$. After a series of steps, the function $f(x) = x_2 + \cos(x_3)$ is produced for the grammar of Figure 1. The production steps are shown in Table 1.

Table 1. A series of production steps for the example chromosome.

String	Chromosome	Operation
<expr>	9,8,6,4,16,10,17,23,8,14	9 mod 3 = 0
(<expr><op><expr>)	8,6,4,16,10,17,23,8,14	8 mod 3 = 2
(<terminal><op><expr>)	6,4,16,10,17,23,8,14	6 mod 2 = 0
(<xlist><op><expr>)	4,16,10,17,23,8,14	4 mod 3 = 1
(x2<op><expr>)	16,10,17,23,8,14	16 mod 4 = 0
(x2+<expr>)	10,17,23,8,14	10 mod 3 = 1
(x2+<func>(<expr>))	17,23,8,14	17 mod 4 = 1
(x2+cos(<expr>))	23,8,14	23 mod 2 = 2
(x2+cos(<terminal>))	8,14	8 mod 2 = 0
(x2+cos(<xlist>))	14	14 mod 3 = 2
(x2+cos(x3))		

2.3. The feature construction method

The technique of feature construction produces artificial features for classification or regression problems as non - linear transformations of the original ones. The new features are evaluated using a machine learning technique, such as artificial neural networks [3,4] or a Radial Basis Function (RBF) networks [11,12]. This method was presented initially in [5]. Also, this method was used in many areas, such as Spam Identification [6], Fetal heart classification [7], signal processing [8,9] etc. The process has the following steps:

1. Initialization step.

- (a) **Obtain** the train data for the current problem. The train data have M pairs (x_i, t_i) , $i = 1..M$. The dimension of each vector x_i is d . The values t_i are the expected outputs for each pattern.
- (b) **Define** the parameters of the genetic algorithm: N_g stands for the the number of allowed generations, N_c represents the number chromosomes, p_s defines the selection rate and p_m the mutation rate.
- (c) **Set** as N_f the number of features that will construct the Grammatical Evolution procedure.
- (d) **Initialize** the chromosomes in the population. Each chromosome is considered as a set of randomly chosen positive integers.
- (e) **Set** $k = 1$, as the generation counter.

2. Genetic step

- (a) **For** $i = 1, \dots, N_c$ **do**
 - i. **Produce** using the Grammatical Evolution procedure a set of N_f artificial features for the corresponding chromosome g_i . These features are produced using the grammar of Figure 1.
 - ii. **Transform** the original dataset using the previously constructed features. The new training set is denoted as $(x_{g_i,j}, t_j)$, $j = 1, ..M$
 - iii. **Train** a machine learning model denoted as C using the new training set. The fitness f_i for the corresponding chromosome is defined as:

$$f_i = \sum_{j=1}^M (C(x_{g_i,j}) - t_j)^2 \quad (1)$$

In the proposed method the RBF network is used as the machine learning model, since the time required for its training is significantly lower than other models, such as neural networks.

- iv. **Perform** the selection procedure. During this procedure the chromosomes are firstly sorted according to their fitness values and the best $(1 - p_s) \times N_C$ of them are copied intact to the next generation. The rest of the chromosomes will be substituted by new chromosomes produced during the procedures of crossover and mutation. 78
79
80
81
82
 - v. **Perform** the crossover procedure. During this procedure $p_s \times N_c$ new chromosomes are produced. For every pair of (\tilde{z}, \tilde{w}) of new chromosomes, two two distinct chromosomes (z, w) are selected from the current population. The selection is performed using the tournament selection. The new chromosomes are produced from the old ones using the one - point crossover procedure, which is graphically outlined in Figure 2. 83
84
85
86
87
88
89
 - vi. **(OK)Perform** the mutation procedure: for each each element of each chromosome, a random number $r \in [0, 1]$ is selected. The underlying element is altered randomly when $r \leq p_m$. 90
91
92
- (b) **EndFor** 93
3. **Set** $k = k + 1$ 94
 4. **If** $k \leq N_G$ goto **Genetic Step**, else terminate the process and obtain the best chromosome g^* with the lowest fitness value. 95
96
 5. **Transform** the corresponding test set, using the N_f features obtained for chromosome g^* . 97
98
 6. **Apply** a machine learning model to the constructed test set and report the associated test error.. 99
100

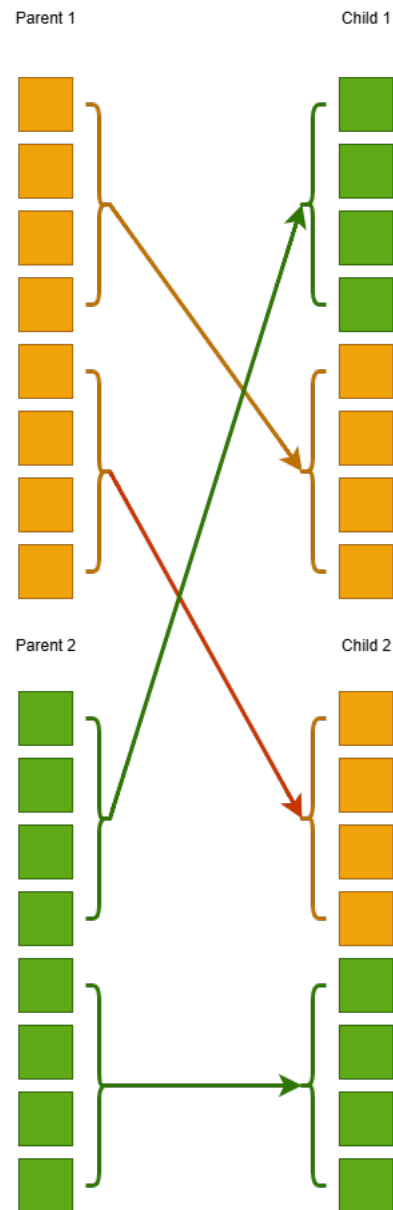


Figure 2. One point crossover method, used in the Grammatical Evolution procedure.

3. Results

The code used in the experiments was code in the C++ programming language and for the optimization methods the freely available Optimus programming tool was incorporated [10]. The experiments were conducted on machine with 128GB of ram, running the Debian Linux operating system. Each experiment was executed 30 times and the average classification error was measured and depicted in the related tables and graphs. Also, the 10 - fold cross validation technique was incorporated for the validation of the experimental results. The values for the parameters of the proposed method are depicted in Table 2.

Table 2. The values for the parameters for the current work.

PARAMETER	MEANING	VALUE
N_g	Number of maximum allowed generations.	500
N_c	Number of chromosomes	500
p_s	Selection rate	0.10
p_m	Mutation rate	0.05
H	Number of processing nodes for neural network	10

In the experimental tables the following notation is used:

1. The column DATASET represents the used dataset.
2. The column RBF stands for the application of an RBF neural network [11,12] with 10 processing nodes on the corresponding dataset.
3. The column GEN represents the application of a genetic algorithm [13] on the training process of a neural network with 10 processing nodes.
4. The column PCA stands for the application of the PCA method [14–16] to construct two artificial features from the original ones. Afterwards, a neural network with 10 processing nodes trained using the BFGS method is applied on the new datasets.
5. The column NNC stands for the application of a neural network constructed with Grammatical Evolution [17] on the corresponding dataset.
6. The column GENCLASS represents the usage of a method that constructs classification rules using Grammatical Evolution [18].
7. The column FC2 is used to represent the application of a genetic algorithm to train a neural network on the dataset produced by the construction of two artificial features using the feature construction method.
8. The column FC3 is used to represent the application of a genetic algorithm to train a neural network on the dataset produced by the construction of three artificial features using the feature construction method.
9. The column FC4 is used to represent the application of a genetic algorithm to train a neural network on the dataset produced by the construction of four artificial features using the feature construction method.
10. The row AVERAGE represents the average classification error for all datasets and the corresponding method.

Table 3. Experimental results for various exercises.

DATASET	RBF	GEN	PCA	NNC	GENCLASS	FC2	FC3	FC4
EXERCISE_0	40.86%	39.63%	44.70%	32.57%	25.52%	11.39%	11.06%	10.35%
EXERCISE 1	38.65%	47.61%	47.35%	42.07%	29.98%	20.16%	16.08%	14.84%
EXERCISE 2	37.57%	35.33%	40.46%	35.01%	31.34%	22.50%	19.78%	22.10%
EXERCISE 3	41.64%	39.17%	43.46%	35.88%	29.00%	19.79%	17.39%	19.91%

Table 4. Precision values.

DATASET	RBF	GEN	PCA	NNC	GENCLASS	FC2	FC3	FC4
EXERCISE_0	58.93%	60.30%	55.89%	67.34%	76.09%	88.51%	88.87%	89.82%
EXERCISE 1	61.99%	49.57%	58.12%	59.48%	71.27%	79.45%	83.66%	85.10%
EXERCISE 2	61.61%	72.58%	58.21%	64.62%	69.02%	77.78%	80.27%	78.01%
EXERCISE 3	58.33%	60.80%	56.63%	64.22%	71.52%	80.29%	82.88%	80.30%

Table 5. Recall values.

DATASET	RBF	GEN	PCA	NNC	GENCLASS	FC2	FC3	FC4
EXERCISE_0	58.84%	60.27%	55.85%	68.92%	74.99%	88.25%	88.49%	89.27%
EXERCISE 1	61.90%	76.55%	60.45%	62.56%	71.11%	79.74%	83.92%	85.42%
EXERCISE 2	62.55%	64.58%	58.94%	65.32%	67.42%	77.06%	80.14%	78.04%
EXERCISE 3	58.30%	60.76%	56.67%	65.19%	71.27%	80.12%	82.65%	80.21%

4. Conclusions

Author Contributions: Fo

Funding: This research has been financed by the European Union : Next Generation EU through the Program Greece 2.0 National Recovery and Resilience Plan , under the call RESEARCH – CREATE

– INNOVATE, project name “iCREW: Intelligent small craft simulator for advanced crew training using Virtual Reality techniques” (project code:TAEDK-06195).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. J. W. Backus. The Syntax and Semantics of the Proposed International Algebraic Language of the Zurich ACM-GAMM Conference. Proceedings of the International Conference on Information Processing, UNESCO, 1959, pp.125-132.
2. M. O’Neill, C. Ryan, Grammatical evolution, IEEE Trans. Evol. Comput. 5, pp. 349–358, 2001.
3. C. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
4. G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of Control Signals and Systems 2, pp. 303-314, 1989.
5. Dimitris Gavrilis, Ioannis G. Tsoulos, Evangelos Dermatas, Selecting and constructing features using grammatical evolution, Pattern Recognition Letters 29, pp. 1358-1365, 2008.
6. Dimitris Gavrilis, Ioannis G. Tsoulos, Evangelos Dermatas, Neural Recognition and Genetic Features Selection for Robust Detection of E-Mail Spam, Advances in Artificial Intelligence Volume 3955 of the series Lecture Notes in Computer Science pp 498-501, 2006.
7. George Georgoulas, Dimitris Gavrilis, Ioannis G. Tsoulos, Chrysostomos Stylios, João Bernardes, Peter P. Groumpos, Novel approach for fetal heart rate classification introducing grammatical evolution, Biomedical Signal Processing and Control 2, pp. 69-79, 2007.
8. Otis Smart, Ioannis G. Tsoulos, Dimitris Gavrilis, George Georgoulas, Grammatical evolution for features of epileptic oscillations in clinical intracranial electroencephalograms, Expert Systems with Applications 38, pp. 9991-9999, 2011.
9. A. T. Tzallas, I. Tsoulos, M. G. Tsipouras, N. Giannakeas, I. Androulidakis and E. Zaitseva, Classification of EEG signals using feature creation produced by grammatical evolution, In: 24th Telecommunications Forum (TELFOR), pp. 1-4, 2016.
10. I.G. Tsoulos, V. Charilogis, G. Kyrou, V.N. Stavrou, A. Tzallas, Journal of Open Source Software 10, 7584, 2025.
11. J. Park and I. W. Sandberg, Universal Approximation Using Radial-Basis-Function Networks, Neural Computation 3, pp. 246-257, 1991.
12. G.A. Montazer, D. Giveki, M. Karami, H. Rastegar, Radial basis function neural networks: A review. Comput. Rev. J 1, pp. 52-74, 2018.
13. Reynolds, J., Rezgui, Y., Kwan, A., & Piriou, S. (2018). A zone-level, building energy optimisation combining an artificial neural network, a genetic algorithm, and model predictive control. Energy, 151, 729-739.
14. Burcu Erkmen, Tülay Yıldırım, Improving classification performance of sonar targets by applying general regression neural network with PCA, Expert Systems with Applications 35, pp. 472-475, 2008.
15. Jing Zhou, Aihuang Guo, Branko Celler, Steven Su, Fault detection and identification spanning multiple processes by integrating PCA with neural network, Applied Soft Computing 14, pp. 4-11, 2014.
16. Ravi Kumar G., Nagamani K., Anjan Babu G., A Framework of Dimensionality Reduction Utilizing PCA for Neural Network Prediction. In: Borah S., Emilia Balas V., Polkowski Z. (eds) Advances in Data Science and Management. Lecture Notes on Data Engineering and Communications Technologies, vol 37. Springer, Singapore. 2020.
17. I.G. Tsoulos, D. Gavrilis, E. Glavas, Neural network construction and training using grammatical evolution, Neurocomputing 72, pp. 269-277, 2008.
18. Nikolaos Anastasopoulos and Ioannis G. Tsoulos and Alexandros Tzallas, GenClass: A parallel tool for data classification based on Grammatical Evolution, SoftwareX, 16, 100830, 2021.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.