Article

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

Aimilios Psathas[1], Ioannis G. Tsoulos[2,*], Nikolaos Giannakeas[3], Alexandros Tzallas[4] and Vasileios Charilogis[5]

[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 in 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.; Giannakeas, N.; 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/

1. Introduction

Global health systems face a significant and growing challenge in the form of neurodegenerative diseases. These conditions cause severe motor and cognitive impairment and are characterized by a progressive disruption of the structure and function of neurons. As the world’s population ages, neurodegenerative diseases like Parkinson’s and Alzheimer’s disease are becoming more common. These conditions have a significant negative impact on patients’ quality of life and are becoming more and more expensive. Innovative methods of early detection, assessment, and treatment are required for these difficult disorders.

A common neurodegenerative disease that primarily affects the motor system is Parkinson’s disease (PD). Tremor, bradykinesia (slow movement), rigidity, and postural instability are classic clinical presentations. Non-motor symptoms like mood disorders, sleep disorders, and cognitive impairment are frequently present as well. Since there are no particular biological tests that can result in an early diagnosis, PD diagnosis primarily depends on clinical evaluation, despite the fact that the condition's symptom profile is complex. As a result, slow symptom progression leads to misdiagnosis or delay, which leads to ineffective treatment, poor patient outcomes, and a lower quality of life [1].

The early detection and monitoring of Parkinson's disease (PD) could be greatly aided by recent technological advancements, particularly in the fields of machine learning (ML) and the internet of things (IoT). IoT sensors such as smartwatches, wearable health monitors, and others allow continuous and non-intrusive monitoring of motor activities, tremor frequencies, and gait abnormalities in real-time. These tools enable an evaluation of a patient's health while they are engaging in their regular activities and continue to gather large amounts of data outside of the typical clinical settings. One can more accurately identify early indicators of Parkinson's disease and enable early interventions by using machine learning techniques on this data [2,3]. For example, ML algorithms can analyze speech patterns, motor skills, and other non-motor symptoms and differentiate subtle patterns that could foretell the full-blown expression of the disease.

The pursuit of accurate, quantitative, and real-time measurement of Parkinson's disease (PD) motor symptoms has been the prime interest of computational neurology. It is rich in the current state-of-the-art research that utilizes diverse sensor modalities and machine learning (ML) methodologies. Pioneering work has validated the potential of smartphone sensor-derived composite scores, like the mobile Parkinson Disease Score (mPDS), that highly correlate to the conventional standards, such as the MDS-UPDRS, and that are capable of tracking intraday symptom variation as well as response to therapy [4]. In the same line, smartphone-enabled active tests as well as passive monitoring during clinical trial protocols have been shown to be realistic, reliable, and highly sensible, usually uncovering abnormalities even among those patients rated as normal on specific items of the UPDRS during the clinic evaluation [5].

In addition to smartphones, specific PD symptoms have been targeted with special-purpose sensors. Quantification of bradykinesia, the PD hallmark, has been achieved successfully with gyrosensors during tapping tests, demonstrating strong correlations with clinical ratings [6]. Tremor measurement has been automated with body-worn accelerometers combined with Hidden Markov Models, classifying tremor type and severity reliably [7]. Moreover, inertial sensors have been used to detect and classify complex gait disorders such as Freezing of Gait (FOG) reliably [8]. Even speech has been targeted in the remote monitoring paradigm, where off-the-shelf speech tests analyzed with signal analysis combined with regression models are capable of tracking PD progression with clinically helpful accuracy [9]. Notably, incorporation of the resulting objective measures into standard care has been shown to improve significantly the resulting clinical outcomes, including the UPDRS as well as quality-of-life scores [10]. Latest advances still investigate the integration of sensor data and advanced ML. Research has looked into the integration of speech signals along with ensemble classifiers such as AdaBoost [11] as well as the processing of gait as well as tremor data obtained from off-the-shelf wearable devices. Still missing, though, is an integrative, high-fidelity method for tracking the subtle kinematics of hand movement, a key domain for the measurement of bradykinesia as well as tremor—beyond controlled smartphone tasks.

This work contributes to this vibrant body of research through the proposal of a unifying methodology grounded in a bespoke SmartGlove system that would allow in-depth analysis of hand movement. Our contribution is tripartite:

- Innovation and pilot use of the SmartGlove, a sensor- rich tool to capture high-accuracy, multi-parameter kinematic output of the hand.

- Construction of a holistic, multiple-component feature selection framework that goes beyond single technologies by integrating statistical significance (t-tests), ensemble learning (Random Forest), and analysis of variance (PCA) to pick the most discriminating, non-redundant biomarkers.
- The recognition of essential attributes—especially non-linear dynamics and spectral traits—that are notably proficient in differentiating motor states in Parkinson's Disease patients both prior to and following medication, offers an enhanced array of digital biomarkers for prospective diagnostic and monitoring applications.

By emphasizing the rich data space of hand kinematics and adopting a rigorous, multi-faceted analytical framework, this research hopes to improve the accuracy and completeness of objective PD evaluation. Briefly stated, the key innovation of this work lies in the combination of a specially crafted data-capture device with an upper-level analytical framework. Contrast this with previous work, which has employed repurposed consumer devices or targeted a very limited set of symptoms, and our specially developed SmartGlove presents a previously unmatched, high-fidelity view of hand motion's rich kinematics. Coupled with our multi-modal feature choice methodology, this system doesn't merely replicate clinical scores but uncovers a novel class of physiologically enlightening biomarkers—specifically in the non-linear dynamic range—that are hidden under conventional analysis. This end-to-end machine learning model is a significant advancement toward precise, data-based neurology from correlation against clinical scales to the detection of underlying digital biomarkers for Parkinson's disease.

In this work, extensive use was made of the Grammatical Evolution [12] technique to create artificial features from existing ones in order to significantly reduce the number of required inputs and to detect hidden correlations in the initial features of the patterns. The applications of the Grammatical Evolution includes cases such as function approximation[13,14], economic problems [15], network security issues [16], water quality problems [17], medical problems [18], evolutionary computing [19], temperature prediction in data centers [20], solving trigonometric problems [21], composing music [22], construction of neural networks [23,24], numerical problems [25], video games [26,27], energy issues [28], combinatorial optimization [29], cryptography [30], production of decision trees [31], problems that appear in electronic circuits [32] etc.

The remainder of this paper follows this organization: Section 2 outlines the materials and methods utilized with a detailed description of the system architecture of the SmartGlove system, the process of data acquisition related to individual exercises of the hands, and the overall feature extraction process. Section 3 details the multi-faceted feature choice strategy and the composite scoring system used to determine the best biomarkers. Finally, Section 4 discusses implications of the research results, points out limitations of the study, and suggests directions of future research.

2. Materials and Methods

This section describes the complete methodology developed for the objective measurement of motor symptoms for Parkinson's Disease (PD). The approach integrates a custom-developed data acquisition system with an advanced computational protocol for conveting raw sensor data into highly discriminative biomarkers. The analysis that follows follows this outline: first, the SmartGlove system and data collection protocol used during special motor experiments are described; second, the multi-domain feature extraction approach is outlined; third, the multi-method scoring regimen developed for feature selection is presented; and, last, the keynote technique of Grammatical Evolution for synthesizing virtual features for improved classification performance, dimension reduction, and generalizability is presented.

2.1. SmartGlove System and Sources of Data

The system of interest here as a primary data source is the SmartGlove system, a specially developed wearable system that has been created with the intent of telemonitoring motor symptoms of PD patients. It has been created with the support of the Operational Program of the Epirus Region 2014-2020 and ESPA 2014-2020 with a view of increasing innovative health and biotechnology solutions. The system of SmartGlove has three primary subsystems:

1. Textile Glove: Made of durable, biocompatible materials, the glove has conductive areas on the palm and fingers that aid movement recognition. It is sweat-resistant and can be washed. It is also expected to last longer than 10 years.
2. System on Chip (SoC): The central processing unit encompasses a low-energy microcontroller that incorporates a Bluetooth 5.0 module, facilitating wireless data communication while utilizing minimal energy resources (with a typical operational range of 10-100 meters, contingent upon the surrounding environment).
3. Sensors: The glove contains a multi-modal sensor suite:
4. Flex sensors to measure finger bending.
5. Contact sensors to detect finger-to-palm interaction.
6. A 9-axis Inertial Measurement Unit (IMU), composed of a 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer, to identify hands' orientation and movement. IMU has outputs of up to a sampling rate of 100 Hz and a resolution of 16 bits.

The system is supplemented with a power management circuitry developed for lithium-polymer batteries, as well as with flash memory used for transient storage of data. The SmartGlove complies with the IEC 60601-1 medical electrical apparatus safety standard, thus ensuring reliability for medical applications. The SmartGlove sends data via Bluetooth to the mobPark mobile app supporting Android 6.0+ as well as iOS 11+. The app acts as an intermediary providing the possibility of managing the profile of the users, providing screen-level cues of exercises, as well as capturing sensor data at a sampling rate of 100 Hz. Then the captured data gets securely sent via an encrypted HTTPS connection to a centralized cloud platform like AWS or Azure for storage, processing, and analyzing purposes with a consideration of the General Data Protection Regulation (GDPR).

2.2. Data Collection Exercises

The data recording at 100 Hz sampling frequency was selected for efficient tremor frequency (4-6 Hz) and bradykinesia capture. Prior to and after the administration of medicines, measurements were recorded, in order for this therapeutic effect of medicines. to be investigated. Four motor activities, according to standardized clinical examination, were performed:

- Exercise 0 (**Resting Tremor Observation)**): The patient was asked to sit with hands on their thighs, palms facing upwards, counting from 1 to 10. This evaluates postural stability, rest tremor. An example of this exercise is shown in Figure 1.
- Exercise 1 (**Test for Postural Tremor and Coordination**): The patient as shown in Figure 2 held hands at shoulder height with fingers interlocked and attempted touching their index fingers together on counting to ten. This evaluates postural tremor and coordination.
- Exercise 2 (**Finger Tapping Speed Test**): Lifting the palm, the subject made ten fast, repeated taps using the index finger on the thumb as shown in Figure 3. This is a routine measure for assessing motor speed and bradykinesia.
- Exercise 3 (**Hand Opening-Closing Test**): The patient did ten fast, sustained cycles of full fist then reopening the hand from a fully stretched palm position. This exercise

evaluates hand rigidity and bradykinesia and an example of this exercise is shown in Figure 4.

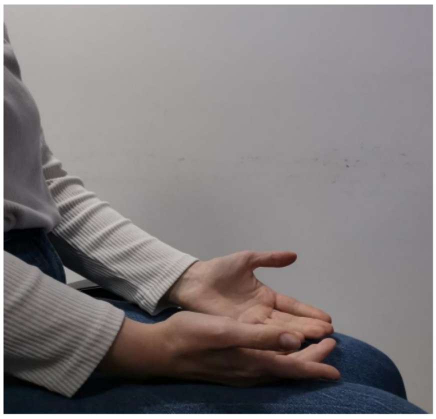

Figure 1. Resting Tremor Observation

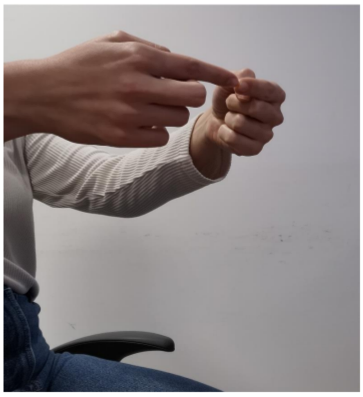

Figure 2. Test for Postural Tremor and Coordination

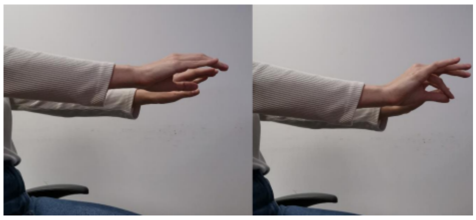

Figure 3. Finger Tapping Speed Test

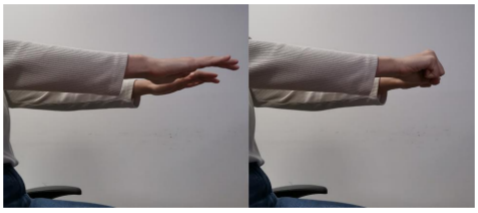

Figure 4. Hand Opening-Closing Test

2.3. Feature extraction

Numerous features were taken from the raw sensor data (accelerometer, gyroscope, magnetometer) for each exercise in order to quantitatively describe the motor symptoms of Parkinson's disease. These features were computed using 50% overlap (50 samples, 0.5

seconds) and sliding windows of 100 samples (1 second). These features extracted are as follows:

- Statistical Properties: These characteristics provide a succinct explanation of the signal's variation and distribution. The following properties were calculated: Skewness, Kurtosis, Quartile Deviation, Mean, Standard Deviation, Variance, Minimum, Maximum, Range, Median, and Interquartile Range (IQR).
- Energy Characteristics: These metrics assess the signal's intensity and degree of activity. Signal Magnitude Area (SMA), Root Mean Square (RMS), Total Energy, and Logarithmic Energy are among the features that were extracted.
- Frequency-Domain Features: These are essential for detecting tremors and take into account the signal's spectral characteristics. Included are the following features: Dominant Frequency, Spectral Flatness, Spectral Flux, Spectral Variability, Spectral Entropy, Spectral Centroid, Spectral Spread, and Spectral Roll-on(85%).
- Dynamic and Nonlinear Features: These characteristics specify the signal's temporal fluctuations, complexity, and predictability. The following characteristics are taken into consideration: Mean Absolute Deviation (MAD), Root Mean Square of Successive Differences (RMSSD), Higuchi Fractal Dimension, Lyapunov Exponent, and Sample Entropy.

Since each feature includes specific aspects of motor impairment typical of the disease, such as tremor regularity, movement amplitude, and signal complexity, the selection of features was driven by a large body of literature on PD analysis. The characteristics for Movement Analysis in Parkinson's Disease are outlined in Table 1.

Table 1. Table of Characteristics for Movement Analysis in Parkinson's Disease.

Category	Characteristics	Purpose	Clinical Significance
Central Tendency	Mean, Median	Median Measures the central value of the signal	Detection of bradykinesia (slowed movements)
Dispersion	Standard deviation, Variance, IQR, QD	Quantifies the spread of data	Identifies motion variability (e.g. gravity, tremor)
Range	Minimum/Maximum value, Range	Records extreme values	Assessment of range of motion reduction in patients with Parkinson's disease
Distribution Shape	Skewness, Kurtosis	Describes asymmetry/peakedness of distribution	Correlation with irregular motor patterns
Variability	Coefficient of Variation (CV), MAD, RMSSD	Normalized measures of dispersion	Variability differentiation before after the treatment
Energy	Total energy, Absolute Energy, RMS	Measures signal intensity	Correlation with hypokinesia (reduced motor energy)
Logarithmic Energy	Log Energy	Enhances subtle energy changes	Detection of small changes
Spectral	Spectral Entropy, Centroid	Analyzes frequency distribution	Localization of tremor and rhythmic abnormality
Roll-off	85% Roll-off	Frequency band where 85% of the power is concentrated	Characterization of tremor bandwidth
Dominant Frequency	Dominant Frequency	Identification of peak frequency	Detection of Parkinson's tremor (4-6Hz)
Spectral Shape	Flatness, Flux, Variability, Dispersion	Quantifies the stability of the spectrum	Unstable tremors vs. rhythmic movements
Dynamics	Lyapunov Exponent, Sampling Entropy	Evaluates the chaos/regularity of the system	Degeneration of motor control in Parkinson's
Fractal	Dimension Higuchi	Measures the complexity of the signal	Loss of fine motor control

2.4. Feature Selection Methodology

Feature selection is a critical step in developing accurate and interpretable machine learning (ML) models for Parkinson's disease (PD) detection. In this study, a multi-method scoring approach was applied to identify the most informative features that differentiate between pre-medication and post-medication states. Instead of relying on a single method, three complementary techniques were employed, and their outputs were combined into a composite score.

2.4.1. Multi-Method Scoring Approach

To accurately evaluate the value of each characteristic for differentiating between pre- and post-medication states, a multi-angled scoring approach was adopted. For each feature, scores for three different aspects, such as statistical significance, importance based on the model, and variance contribution, were computed.

- Statistical Significance (Paired T-Test): The paired T-Test was then carried out on measurements preceding and succeeding medication on each dimension. To make the significance a number, the p_{value} was transformed into a score using the following formula:

$$\text{score}_{T_{\text{test}}} = -\log_{10}(p_{\text{value}} + \epsilon) \tag{1}$$

  Here, $e = 1 \times 10^{-10}$ is a very small constant introduced for avoiding numerical instability arising from a p_{value} of zero. This mapping makes variables with lower p_{value} have exponentially large scores, hence indicating more considerable statistical evidence against the null hypothesis ([41].
- **Model-Based Importance** (**Random Fores**t): The Random Forest classifier was built such that it could distinguish between the two different states. The importance of a specific feature was computed using the average reduction in Gini impurity across all trees in the ensemble. Gini impurity is a measure of the purity of a node, and the importance is given by the weighted cumulative reduction in node impurity, divided by the probability of the node, and averaged across trees [42]. The evaluation score_{RF} for a feature, f is given by:

$$\text{score}_{RF} = \frac{1}{N_{\text{trees}}} \sum \tag{2}$$

  where N_{trees} is the overall number of trees, T is one tree, and $\Delta\text{Gini}(n, f)$ decrease in Gini impurity at node n when a split is made along feature f. High-scoring variables have a larger effect on the model predicting **(Breiman, 2001).**
- **Variance Contribution Score** (**PCA**): Principal Component Analysis (PCA) was performed on the standardized data set. The contribution of a individual original feature towards the principal components (PCs) was measured by adding its absolute loadings across all PCs. The loading of a feature onto a PC is its correlation with the specific component [43]. The score $\text{score}_{\text{PCA}}$ was computed as follows:

$$\text{score}_{\text{PCA}} = \sum_i |\text{loading}_i| \tag{3}$$

  Here, the loading_i is the feature weight corresponding to the i-th principal component. This cumulative sum provides the total impact of the feature on the total variance tackled by the Principal Component Analysis (PCA) model **(Jolliffe & Cadima, 2016).**

2.4.2. Composite Score Calculation

To reduce these unrelated measurements into one unified measure of feature prominence, the scores were amalgamated. Each score was first normalized between $[0, 1]$ using min-max normalization for comparability across scores. A clear composite score was then calculated as a weighted average of the normalized scores:

$$\text{Composite Score} = 0.4 \times \text{score}_{T_{\text{test}}} + 0.3 \times \text{score}_{RF} + 0.3 \times \text{score}_{\text{PCA}} \tag{4}$$

This weighting approach was chosen so that features which are highly statistically significant (by the T-test) are given priority, first, while data from model-based learning (Random Forest) and variance-based analysis (PCA) are incorporated cautiously

2.4.3. Selection of best features

Features were ranked according to their composite score, and the top-performing ones were selected for subsequent model development. Representative high-ranking features includes:

- RMSSD from the gyroscope during Exercise 2 (score: 0.663),
- Lyapunov Exponent (score: 0.644),
- Higuchi Fractal Dimension (score: 0.612).

These selected features served as the foundation for later predictive modeling and analysis, aiming to capture the most relevant motor signal characteristics associated with PD symptoms [44,45]. In Table 2 the top 20 features are presented.

Table 2. Top 20 features scored

Exercise	Sensor	Feature Name	Composite Score
2	gyro	rmssd	0.663
2	gyro	lyapunov_exponent	0.644
2	gyro	higuchi_fractal_dimension	0.612
2	gyro	range	0.573
3	acce	higuchi_fractal_dimension	0.469
3	gyro	max	0.452
3	acce	lyapunov_exponent	0.437
2	gyro	spectral_rolloff	0.433
2	acce	sample_entropy	0.422
0	magn	lyapunov_exponent	0.406
0	magn	spectral_variability	0.399
0	magn	higuchi_fractal_dimension	0.399
0	magn	range	0.381
3	magn	rmssd	0.376
0	gyro	rmssd	0.375
2	gyro	std	0.373
3	acce	spectral_flux	0.367
1	gyro	lyapunov_exponent	0.365
1	gyro	rmssd	0.361
1	gyro	max	0.358

2.5. Grammatical Evolution preliminaries

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 [33]. 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 $S \in N$.
- 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 5. The notation $<>$ 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 Rule = V mod 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 5. The production steps are shown in Table 3.

Figure 5. 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)
```

Table 3. 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))		

2.6. 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 [34,35] or a Radial Basis Function (RBF) networks [47,48]. This method was presented initially in [36]. Also, this method was used in many areas, such as Spam Identification [37], Fetal heart classification [38], signal processing [39,40] 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** $i = 1, \ldots, N_c$ **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 5.
      ii. **Transform** the original dataset using the previously constructed features. The new training set is denoted as $\left(x_{g_i,j}, t_j\right)$, $j = 1, ..M$
      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:

$$f_i = \sum_{j=1}^{M} \left(C\left(x_{g_i,j}\right) - t_j\right)^2 \quad (5)$$

      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.

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 6.

vi. **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 $r \leq p_m$.

(b) **EndFor**

3. **Set** $k = k + 1$
4. **If** $k \leq N_G$ goto **Genetic** Step, else terminate the process and obtain the best chromosome g^* with the lowest fitness value.
5. **Transform** the corresponding test set, using the N_f features obtained for chromosome g^*.
6. **Apply** a machine learning model to the constructed test set and report the associated test error..

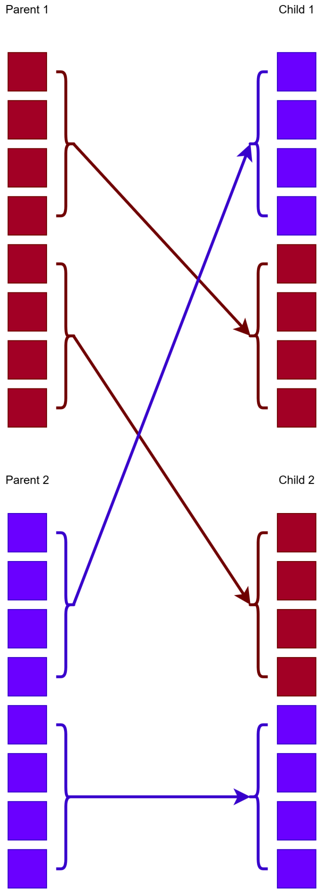

Figure 6. One point crossover method, used in the Grammatical Evolution procedure.

A flowchart of the previous algorithm is shown in Figure 7.

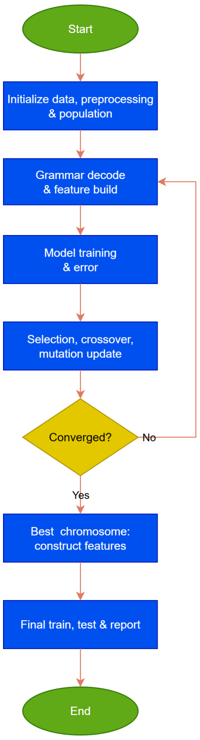
Start
Initialize data, preprocessing & population
Grammar decode & feature build
Model training & error
Selection, crossover, mutation update
Converged?
No
Yes
Best chromosome: construct features
Final train, test & report
End

Figure 7. Flowchart of the used genetic algorithm.

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 [46]. 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 4.

Table 4. 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 performed exercise.
2. The column RBF stands for the application of an RBF neural network [47,48] with 10 processing nodes on the corresponding dataset.
3. The column GEN represents the application of a genetic algorithm [49] on the training process of a neural network with 10 processing nodes.
4. The column PCA stands for the application of the PCA method [50–52] 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 [53] on the corresponding dataset.
6. The column GENCLASS represents the usage of a method that constructs classification rules using Grammatical Evolution [54].
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 5 reports classification error per exercise and per feature/model family, where lower values are better. The proposed Feature Construction approach (columns FC2, FC3 and FC4) consistently yields lower errors than classical baselines such as PCA, RBF, NNC, GEN and GENCLASS, indicating that the constructed descriptors capture the discriminative motion patterns more effectively. Based on the overall average, FC3 is the strongest among the proposed variants, achieving the lowest mean error, with FC4 very close behind and FC2 also competitive. In EXERCISE 0 the Feature Construction variants clearly dominate, with FC3 providing the best balance between expressiveness and generalization, FC4 performing nearly as well, and FC2 remaining stable. In EXERCISE 1 the picture is similar: FC2–FC4 retain a lead over classical pipelines and FC3 acts as the "sweet spot," while FC4 is the slightly more aggressive variant and FC2 the conservative yet reliable option. In EXERCISE 2, where separability is more challenging, the FC columns keep the error noticeably lower than the classical models and FC3 generally maintains an edge, although FC4 occasionally approaches or overlaps within small margins. In EXERCISE 3 the same trend holds, with FC2, FC3 and FC4 achieving lower errors and FC3 preserving the overall lead among the proposed methods. Overall, Table 5 underscores that representation quality precedes classifier complexity; when features are properly constructed, even relatively simple decision rules can achieve consistently low error. In practice, orchestrating Feature Construction with linear or RBF-based decision layersor adopting a hybrid design where feature construction is followed by mild dimensionality reduction preserves the information advantage while further controlling noise. The average results for this table are also outlined graphically in Figure 8.

Table 5. 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%

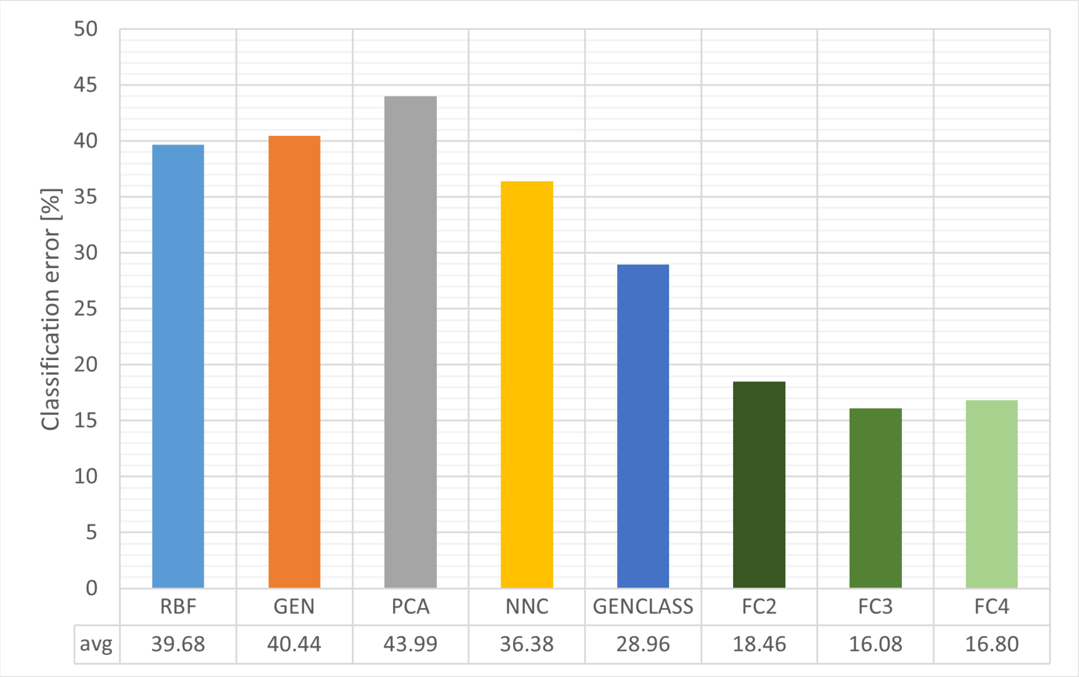

Figure 8. The average classification error for all methods participated in the experiments.

Also the precision values for all exercises and machine learning methods are outlined in Table 6. Similarly, the recall values are depicted in Table 7. 397 398

Table 6. Precision values for all methods participated in the experiments.

DATASET	RBF	GEN	PCA	NNC	GENCLASS	FC2	FC3	FC4
EXERCISE 0	0.589	0.603	0.559	0.673	0.761	0.885	0.889	0.898
EXERCISE 1	0.62	0.496	0.581	0.595	0.713	0.795	0.837	0.851
EXERCISE 2	0.616	0.726	0.582	0.646	0.69	0.778	0.803	0.78
EXERCISE 3	0.583	0.608	0.566	0.642	0.715	0.803	0.829	0.803

Table 7. Recall values for all methods used in the conducted experiments.

DATASET	RBF	GEN	PCA	NNC	GENCLASS	FC2	FC3	FC4
EXERCISE 0	0.588	0.603	0.559	0.689	0.75	0.883	0.885	0.893
EXERCISE 1	0.619	0.766	0.605	0.626	0.711	0.797	0.839	0.854
EXERCISE 2	0.626	0.646	0.589	0.653	0.674	0.771	0.801	0.78
EXERCISE 3	0.583	0.608	0.567	0.652	0.713	0.801	0.827	0.802

4. Conclusions 399

Author Contributions: Fo 400

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). 401 402 403 404

Institutional Review Board Statement: Not applicable. 405

Informed Consent Statement: Not applicable. 406

Conflicts of Interest: The authors declare no conflicts of interest. 407

References 408

1. Jankovic, J., 2008. Parkinson’s disease: clinical features and diagnosis. J Neurol Neurosurg Psychiatry 79, 368. 409

2. Javaid, M., Haleem, A., Pratap Singh, R., Suman, R., Rab, S., 2022. Significance of machine learning in healthcare: Features, pillars and applications. International Journal of Intelligent Networks 3, 58–73.
3. C., Wang, J., Wang, S., Zhang, Y., 2024. A review of IoT applications in healthcare. Neurocomputing 565, 127017.
4. Zhan, A., Mohan, S., Tarolli, C., Schneider, R.B., Adams, J.L., Sharma, S., Elson, M.J., Spear, K.L., Glidden, A.M., Little, M.A., Terzis, A., Dorsey, E.R., Saria, S., 2018. Using Smartphones and Machine Learning to Quantify Parkinson Disease Severity: The Mobile Parkinson Disease Score. JAMA Neurology 75, 876–880.
5. Lipsmeier, F., Taylor, K.I., Kilchenmann, T., Wolf, D., Scotland, A., Schjodt-Eriksen, J., Cheng, W.Y., Fernandez-Garcia, I., Siebourg-Polster, J., Jin, L., Soto, J., Verselis, L., Boess, F., Koller, M., Grundman, M., Monsch, A.U., Postuma, R.B., Ghosh, A., Kremer, T., Czech, C., Gossens, C., Lindemann, M., 2018. Evaluation of smartphone-based testing to generate exploratory outcome measures in a phase 1 Parkinson's disease clinical trial. Movement Disorders 33, 1287–1297.
6. Kim, J.W., Lee, J.H., Kwon, Y., Kim, C.S., Eom, G.M., Koh, S.B., Kwon, D.Y., Park, K.W., 2011. Quantification of bradykinesia during clinical finger taps using a gyrosensor in patients with Parkinson's disease. Medical & Biological Engineering & Computing 49, 365–371.
7. Rigas, G., Gatsios, D., Fotiadis, D.I., Tsakanikas, V., Tsilikis, I., Konitsiotis, S., 2012. Assessment of Tremor Activity in the Parkinson's Disease Using a Set of Wearable Sensors. IEEE Transactions on Information Technology in Biomedicine 16, 478–487.
8. Djuric-Jovicic, M., Jovicic, N.S., Radovanovic, S.M., Stankovic, I.D., Popovic, M.B., Kostic, V.S., 2014. Automatic Identification and Classification of Freezing of Gait Episodes in Parkinson's Disease Patients. IEEE Transactions on Neural Systems and Rehabilitation Engineering 22, 685–694.
9. Tsanas, A., Little, M.A., McSharry, P.E., Ramig, L.O., 2010. Accurate telemonitoring of Parkinson's disease progression by noninvasive speech tests. IEEE Transactions on Biomedical Engineering 57, 884–893.
10. Farzanehfar, P., Woodrow, H., Braybrook, M., McGregor, S., Evans, A., Nicklason, F., Horne, M., 2018. Objective measurement in routine care of people with Parkinson's disease improves outcomes. NPJ Parkinson's Disease 4, 10.
11. Bukhari, S.N.H., Ogudo, K.A., 2024. Ensemble Machine Learning Approach for Parkinson's Disease Detection Using Speech Signals. Mathematics 12, 1575.
12. M. O'Neill, C. Ryan, Grammatical evolution, IEEE Trans. Evol. Comput. **5,**pp. 349–358, 2001.
13. C. Ryan, J. Collins, M. O'Neill, Grammatical evolution: Evolving programs for an arbitrary language. In: Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C. (eds) Genetic Programming. EuroGP 1998. Lecture Notes in Computer Science, vol 1391. Springer, Berlin, Heidelberg, 1998.
14. M. O'Neill, M., C. Ryan, Evolving Multi-line Compilable C Programs. In: Poli, R., Nordin, P., Langdon, W.B., Fogarty, T.C. (eds) Genetic Programming. EuroGP 1999. Lecture Notes in Computer Science, vol 1598. Springer, Berlin, Heidelberg, 1999.
15. A. Brabazon, M. O'Neill, Credit classification using grammatical evolution, Informatica **30.3**, 2006.
16. S. Şen, J.A. Clark. A grammatical evolution approach to intrusion detection on mobile ad hoc networks, In: Proceedings of the second ACM conference on Wireless network security, 2009.
17. L. Chen, C.H. Tan, S.J. Kao, T.S. Wang, Improvement of remote monitoring on water quality in a subtropical reservoir by incorporating grammatical evolution with parallel genetic algorithms into satellite imagery, Water Research **42**, pp. 296-306, 2008.
18. J. I. Hidalgo, J. M. Colmenar, J.L. Risco-Martin, A. Cuesta-Infante, E. Maqueda, M. Botella,J. A. Rubio, Modeling glycemia in humans by means of Grammatical Evolution, Applied Soft Computing **20**, pp. 40-53, 2014.
19. J. Tavares, F.B. Pereira, Automatic Design of Ant Algorithms with Grammatical Evolution. In: Moraglio, A., Silva, S., Krawiec, K., Machado, P., Cotta, C. (eds) Genetic Programming. EuroGP 2012. Lecture Notes in Computer Science, vol 7244. Springer, Berlin, Heidelberg, 2012.
20. M. Zapater, J.L. Risco-Martín, P. Arroba, J.L. Ayala, J.M. Moya, R. Hermida, Runtime data center temperature prediction using Grammatical Evolution techniques, Applied Soft Computing **49**, pp. 94-107, 2016.
21. C. Ryan, M. O'Neill, J.J. Collins, Grammatical evolution: Solving trigonometric identities, proceedings of Mendel. Vol. 98. 1998.
22. A.O. Puente, R. S. Alfonso, M. A. Moreno, Automatic composition of music by means of grammatical evolution, In: APL '02: Proceedings of the 2002 conference on APL: array processing languages: lore, problems, and applications July 2002 Pages 148–155.
23. Lídio Mauro Limade Campo, R. Célio Limã Oliveira,Mauro Roisenberg, Optimization of neural networks through grammatical evolution and a genetic algorithm, Expert Systems with Applications **56**, pp. 368-384, 2016.
24. K. Soltanian, A. Ebnenasir, M. Afsharchi, Modular Grammatical Evolution for the Generation of Artificial Neural Networks, Evolutionary Computation **30**, pp 291–327, 2022.
25. I. Dempsey, M.O' Neill, A. Brabazon, Constant creation in grammatical evolution, International Journal of Innovative Computing and Applications **1** , pp 23–38, 2007.
26. E. Galván-López, J.M. Swafford, M. O'Neill, A. Brabazon, Evolving a Ms. PacMan Controller Using Grammatical Evolution. In: , et al. Applications of Evolutionary Computation. EvoApplications 2010. Lecture Notes in Computer Science, vol 6024. Springer, Berlin, Heidelberg, 2010.

27. N. Shaker, M. Nicolau, G. N. Yannakakis, J. Togelius, M. O'Neill, Evolving levels for Super Mario Bros using grammatical evolution, 2012 IEEE Conference on Computational Intelligence and Games (CIG), 2012, pp. 304-31.
28. D. Martínez-Rodríguez, J. M. Colmenar, J. I. Hidalgo, R.J. Villanueva Micó, S. Salcedo-Sanz, Particle swarm grammatical evolution for energy demand estimation, Energy Science and Engineering **8**, pp. 1068-1079, 2020.
29. N. R. Sabar, M. Ayob, G. Kendall, R. Qu, Grammatical Evolution Hyper-Heuristic for Combinatorial Optimization Problems, IEEE Transactions on Evolutionary Computation **17**, pp. 840-861, 2013.
30. C. Ryan, M. Kshirsagar, G. Vaidya, G. et al. Design of a cryptographically secure pseudo random number generator with grammatical evolution. Sci Rep **12**, 8602, 2022.
31. P.J. Pereira, P. Cortez, R. Mendes, Multi-objective Grammatical Evolution of Decision Trees for Mobile Marketing user conversion prediction, Expert Systems with Applications **168**, 114287, 2021.
32. F. Castejón, E.J. Carmona, Automatic design of analog electronic circuits using grammatical evolution, Applied Soft Computing **62**, pp. 1003-1018, 2018.
33. 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.
34. C. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
35. G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of Control Signals and Systems **2**, pp. 303-314, 1989.
36. Dimitris Gavrilis, Ioannis G. Tsoulos, Evangelos Dermatas, Selecting and constructing features using grammatical evolution, Pattern Recognition Letters **29**,pp. 1358-1365, 2008.
37. 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.
38. 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
39. 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
40. 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.
41. Tukey, J.W., 1977. Exploratory Data Analysis. Addison-Wesley.
42. Patel, S., Lorincz, K., Hughes, R., Huggins, N., Growdon, J., Standaert, D., Akay, M., Dy, J., Welsh, M., Bonato, P., 2009. Monitoring Motor Fluctuations in Patients With Parkinson's Disease Using Wearable Sensors. IEEE Trans. Inform. Technol. Biomed. 13, 864–873.
43. Guyon, I., Elisseeff, A., 2003. An Introduction to Variable and Feature Selection. Journal of Machine Learning Research 3, 1157–1182.
44. Goetz, C.G., Tilley, B.C., Shaftman, S.R., Stebbins, G.T., Fahn, S., Martinez-Martin, P., Poewe, W., Sampaio, C., Stern, M.B., Dodel, R., Dubois, B., Holloway, R., Jankovic, J., Kulisevsky, J., Lang, A.E., Lees, A., Leurgans, S., LeWitt, P.A., Nyenhuis, D., Olanow, C.W., Rascol, O., Schrag, A., Teresi, J.A., Van Hilten, J.J., LaPelle, N., 2008. Movement Disorder Society-sponsored revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS): Scale presentation and clinimetric testing results. Movement Disorders 23, 2129–2170.
45. Chen, Z., Pang, M., Zhao, Z., Li, S., Miao, R., Zhang, Yifan, Feng, Xiaoyue, Feng, Xin, Zhang, Yexian, Duan, M., Huang, L., Zhou, F., 2020. Feature selection may improve deep neural networks for the bioinformatics problems. Bioinformatics 36, 1542–1552.
46. I.G. Tsoulos, V. Charilogis, G. Kyrou, V.N. Stavrou, A. Tzallas, Journal of Open Source Software **10**, 7584, 2025.
47. J. Park and I. W. Sandberg, Universal Approximation Using Radial-Basis-Function Networks, Neural Computation **3**, pp. 246-257, 1991.
48. G.A. Montazer, D. Giveki, M. Karami, H. Rastegar, Radial basis function neural networks: A review. Comput. Rev. J **1**, pp. 52-74, 2018.
49. 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.
50. 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.
51. 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.
52. 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.

53. I.G. Tsoulos, D. Gavrilis, E. Glavas, Neural network construction and training using grammatical evolution, Neurocomputing **72**, pp. 269-277, 2008.
54. 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.