Article

Creating features using particle swarm optimization

Ioannis G. Tsoulos[1,*], Alexandros Tzallas[2]

[1] Department of Informatics and Telecommunications, University of Ioannina, Greece;itsoulos@uoi.gr
[2] Department of Informatics and Telecommunications, University of Ioannina, Greece;tzallas@uoi.gr
* Correspondence: itsoulos@uoi.gr;

Abstract: The problem of data classification or data fitting is widely applicable in a multitude of scientific areas, and for this reason a number of machine learning models have been developed. However, in many cases, these models present problems of overfitting and cannot generalize satisfactorily to unknown data. Furthermore, in many cases, many of the features of the input data do not contribute to learning, or there may even be hidden correlations between the features of the dataset. The current article proposes a method based on Particle Swarm Optimization and Grammatical Evolution, that creates artificial features from the original ones. In addition, this new technique utilizes penalty factors to limit the generated features to a range of values to make training machine learning models more efficient. The method was tested on a series of well - known datasets from the relevant literature and was tested against a series of widely used machine learning models.

Keywords: Particle swarm optimization, Grammatical Evolution, Evolutionary techniques, Stochastic methods.

1. Introduction

A multitude of everyday problems from various sciences can be treated as a problem of categorization or regression problems, such as problems that appear in the fields of physics [1–4], chemistry [5–7], economics [8,9], environmental problems [10–12], medical problems [13,14] etc. In the relevant literature there is a wide range of techniques that one can use to handle such problems such as the k nearest neighboors model (k-NN) [15,16], artificial neural networks (ANNs) [17,18], Radial Basis Function (RBF) networks [19–21], Support Vector Machines (SVM) [22,23], decision trees [24,25] etc. A systematic review of methods used in classification problems can be found in the paper of Kotsiantis et al [26].

In most cases, learning models have a number of parameters that should be determined through some algorithms, such as the Back Propagation method [27,28] for the artificial neural networks or more advanced optimization methods such as the Genetic algorithms [29–31]. However, most of the time there are two main problems in the parameterization of learning models:

- Requirement for large training time, which is proportional to the dimension of the input data. For example, in a neural network with one hidden layer equipped with 10 processing nodes and a provided dataset with 10 inputs, then more than $N = 100$ parameters are required to build the neural network. Therefore the size of the network will grow proportionally to the problem and therefore longer training times will be required for the model. In addition, in some techniques such as for example Bfgs [32], $O\left(N^2\right)$ storage space will be required for the training model and for the partial derivatives required by the optimization method. An extensive discussion of the problems caused by increased data dimensionality is presented in the paper by Verleysen et al [33]. Some common approaches to reduce the dimension of the input datasets are the Principal Component Analysis (PCA) method [34–36] as well as the Minimum redundancy feature selection (MRMR) technique [37,38]. Furthermore, Wang et al proposed an auto - encoder reduction method, applied on a seires of large datasets[39].

Citation: Tsoulos, I.G.; Tzallas A Creating features using particle swarm optimization. *Journal Not Specified* **2022**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2023 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/).

- The problem of reduced performance of models on unknown data, also known as overfitting problem. The paper of Geman et al [40] as well the article of Hawkins [41] thoroughly discuss on the topic of overfitting. Examples of techniques proposed to overcome this problem are the weight sharing methods [42,43], the prunning methods [44–46], weight elimination [47–49], weight decaying methods [50,51] etc.

This article proposes a two phase method to overcome the above problems. During the first phase, a limited number of artificial features are created from the original ones using a method based on the Grammatical Evolution procedure [52]. Grammatical Evolution is an evolutionary process, where the chromosomes are production rules of the target BNF grammar and it has been used sucesfully in a variety of applications, such as music composition [53], economics [54], symbolic regression [55], robotics [56], caching algorithms [57] etc. These features are iteratively adjusted using a hybrid technique based on particle swarm optimization (PSO)[58–60], where the generated features are constrained using penalty factors to be within a pre-defined value interval and their evaluation is done using an RBF network. The RBF network was preferred over other machine learning models to be used to evaluate the generated features due to its training speed. The PSO method has been selected as the optimization method due to its simplicity and the small number of parameters that should be set. Also, the PSO method has been used in many difficult problems from all areas of the sciences, such as problems that arise in physics [61,62], chemistry [63,64], medicine [65,66], economics [67] etc. Furthermore, the PSO method was successfully applied recently in many practical problems such as flow shop scheduling [68], successful development of electric vehicle charging strategies [69], emotion recognition [70], robotics [71] etc.

The idea of creating artificial features using Grammatical Evolution was firstly introduced in the paper of Gavrilis et al [72] and it has been succesfully applied on a series of problems, such as Spam Identification [73], Fetal heart classification [74], epileptic oscillations[75], construction of Covid-19 predictive models [76], performance and early drop prediction for higher education students [77] etc.

Feature selection using neural networks has been also proposed in a series of papers, such as the work of Verikas and Bacauskiene [78] or the work of Kabir et al [79]. Moreover, Devi utlized a Simulated Annealing approach [80] to select the most important features for classification datasets. Also, Neshatian et al [81] developed a genetic algorithm that produces features using an entropy based fitness function.

The rest of this article is divided as follows: in section 2 the steps of the proposed method are fully described, in section 3 the used experimental datasets as well as the results obtained by the incorporation of the proposed method are outlined and finally in section 4 some conclusions are listed.

2. The proposed method

The basic steps by which the Grammatical Evolution technique produces the artificial features are then analyzed as well as the steps of the overall process of creating and evaluating the artificial features.

2.1. The technique of Grammatical Evolution

The process of Grammatical Evolution uses chromosomes that represent production rules of the underlying BNF (Backus–Naur form) grammar[82] of the objective problem. BNF grammars have been widely used to describe the syntax of programming languages. Any BNF grammar is a set $G = (N, T, S, P)$, where

- The set N represents the non-terminal symbols of the grammar. Any non - terminal symbol is analyzed to a series of terminal symbols using the production rules of the grammar.
- T is the set of terminal symbols.
- The non - terminal symbol S represents the start symbol of the grammar.

- The set P contains the production rules of the grammar. Typically, any production rule is expressed in the form $A \to a$ or $A \to aB,\ A, B \in N,\ a \in T$.

The process that creates a valid program, starts from the symbol S and gradually replaces non-terminal symbols with the right hand of the selected production rule from the provided chromosome. The rule is selected with the following steps:

- Get the next element from the chromosome and denote it as V.
- Select the production rule with the scheme Rule = V mod N_R, where N_R is the total number of production rules for the current non – terminal symbol.

The BNF grammar for the proposed method is shown in Figure 1. The constant N is the dimension of the input dataset.

Figure 1. BNF grammar of the proposed method.

```
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)
            |<digitlist>.<digitlist> (1)
<xlist>::=x1    (0)
            | x2 (1)
            .........
            | xN (N)
<digitlist>::=<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)
```

An example that produces a valid expression for the chromosome

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

with $N = 3$ is shown in Table 1. The final expression that created is $f(x) = x_2 + \cos(x_3)$.

Table 1. Steps to produce a valid expression from the BNF grammar.

Expression	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	$23 \mod 2 = 1$
(x2+cos(<terminal>))	8,14	$8 \mod 2 = 0$
(x2+cos(<xlist>))	14	$14 \mod 3 = 2$
(x2+cos(x3))		

2.2. Feature construction

The proposed method is used to created N_f artificial features from the original ones. The new features will be considered as non - linear combinations of the old features and the process for any particle p has as follows:

1. **Divide** p into N_f parts. Every part is denoted as the p_i sub - particle.
2. **For** each sub - particle p_i a new artificial feature $g_i(\overrightarrow{x}, p_i)$ is constructed with the grammar of Figure 1 as a non - linear combination of the original set of features $\overrightarrow{x}$.

The final set of features will be considered as mapping functions of the original ones. For example the set:

$$g(\overrightarrow{x}, p) = \begin{cases} g_1(\overrightarrow{x}, p_1) & = & x_1^2 + 2x_3 \\ g_2(\overrightarrow{x}, p_2) & = & 3\cos(x_2) \end{cases}$$

is a set of mapping functions for the original features $\overrightarrow{x} = (x_1, x_2, x_3)$. However, sometimes the generated features can lead to extreme values and this will result in generalization problems from the used machine learning models. For this reason and in the present work, penalty factors are used so that the mapping functions do not lead to extreme values. These penalty factors also modify the fitness function that the Particle Swarm Optimization technique will minimize each time and are considered next.

2.3. Fitness calculation

The following steps calculates the fitness for any given particle p.

1. **Denote as** $\mathrm{TO} = \left\{(\overrightarrow{x_1}, y_1), (\overrightarrow{x_2}, y_2), \ldots, (\overrightarrow{x_M}, y_M)\right\}$ the original train set.
2. **Set** $V = 0$, the penalty factor
3. **Compute** the mapping function $g(\overrightarrow{x}, p)$ as suggested in subsection 2.2
4. **Set** TF=$\emptyset$,the modified train set
5. **For** $i = 1, \ldots, M$ **do**
   (a) **Set** $\widetilde{x}_i = g(\overrightarrow{x_i}, p)$
   (b) **Set** $\mathrm{TF} = \mathrm{TF} \cup (\widetilde{x}_i, y_i)$
   (c) **If** $\|\widetilde{x}_i\| > L_{\max}$, **then** $V = V + 1$, where $L_{\max}$ a predefined positive value.
6. **End For**
7. **Train** an RBF $C(x)$ with H processing NODES on TF and obtain the following error:

$$f_p = \sum_{j=1}^{M} \left(C(\widetilde{x}_i) - y_j\right)^2 \quad (1)$$

8. **Compute** the final fitness value:

$$f_p = f_p \times \left(1 + \lambda V^2\right) \tag{2}$$

where $\lambda > 0$.

2.4. The proposed PSO method

The mains steps for this algorithm are outlined in detail in Algorithm 1.

Algorithm 1 The base PSO algorithm executed in one processing unit.

---

1. **Initialization Step** .
   (a) **Set** iter $= 0$.
   (b) **Set** m as the total number of particles.
   (c) **Set** iter_{max} as the maximum number of iterations allowed.
   (d) **Initialize** randomly, the positions $p_1, p_2, ..., p_m$ for the particles. For the grammatical evolution, every chromosome is a series of randomly selected integers.
   (e) **Initialize** randomly the velocities $u_1, u_2, ..., u_m$. For the current work every vector of velocities is a series of randomly selected integers in a the range $[u_{\text{min}}, u_{\text{max}}]$. In the current work $u_{\text{min}} = -5$, $u_{\text{max}} = 5$.
   (f) **For** $i = 1..m$ do $b_i = p_i$. The vector b_i denotes the best located position of particle p_i.
   (g) **Set** $p_{\text{best}} = \arg\min_{i \in 1..m} f(p_i)$
2. **Termination Check Step** . If iter $\geq \text{iter}_{\text{max}}$ then goto step 8.
3. **For** $i = 1..m$ **Do**
   (a) **Compute** the velocity u_i as a combination of the vectors u_i, p_i and p_{best}
   (b) **Set** the new position for the particle as: $p_i = p_i + u_i$
   (c) **Calculate** the fitness $f(p_i)$ for particle p_i using the procedure described in subsection 2.3.
   (d) **If** $f(p_i) \leq f(b_i)$ then $b_i = p_i$
4. **End For**
5. **Set** $p_{\text{best}} = \arg\min_{i \in 1..m} f(p_i)$
6. **Set** iter $=$ iter $+ 1$.
7. **Goto** Step 2
8. **Test step**. Apply the mapping function of the best particle p_{best} to the test set of the problem and apply a machine learning model obtaining the corresponding test error.

The above calculates at every iteration the new position of the particle i using:

$$p_i = p_i + u_i \tag{3}$$

In most cases the new velocity could be a linear combination of the previously computed velocity and the best values b_i and p_{best} and it can be defined as:

$$u_i = \omega u_i + r_1 c_1 (b_i - p_i) + r_2 c_2 (p_{\text{best}} - p_i) \tag{4}$$

where

1. The variables r_1, r_2 are random numbers defined in $[0, 1]$.
2. The constants c_1, c_2 are defined in range $[1, 2]$.
3. The variable ω, commnly called inertia, was suggested by Shi and Eberhart [58] and in the current work was computer through the following equation

$$\omega_{\text{iter}} = 0.5 + \frac{r}{2} \tag{5}$$

The variable r is a a random number with $r \in [0,1]$.

3. Experiments

The ability of the proposed technique to produce effective artificial features for class prediction and feature learning will be measured in this section on a range of datasets from the relevant literature. This data comes from the relevant websites:

1. UCI dataset repository, https://archive.ics.uci.edu/ml/index.php
2. Keel repository, https://sci2s.ugr.es/keel/datasets.php[83].
3. The Statlib URL ftp://lib.stat.cmu.edu/datasets/index.html.

The proposed technique will be compared with a series of known machine learning techniques and the experimental results are then presented in the relevant tables.

3.1. Experimental datasets

The classification problems used in the experiments have as follows:

1. **Appendictis**, a medical dataset that represents 7 medical measures for 106 patients on which the class label represents if the patient has appendicitis [84,85].
2. **Australian** dataset [86], an dataset concerning economical transactions in banks.
3. **Balance** dataset, a dataset generated to model psychological experimental results[87].
4. **Bands** dataset, a dataset used in rotogravure printing [88].
5. **Dermatology** dataset [89], a medical dataset used to detect the type of Eryhemato-Squamous Disease.
6. **Hayes roth** dataset [91].
7. **Heart** dataset [90], a mecical dataset used to detect heart diseases.
8. **HouseVotes** dataset [92], a dataset related to the conressional voting records of USA.
9. **Ionosphere** dataset, used to classify measurements from the ionosphere and it has been examined in a variety of research papers [93,94].
10. **Liverdisorder** dataset [95,96], a medical dataset.
11. **Mammographic** dataset [97], a medical dataset used for breast cancer diagnosis.
12. **Parkinsons** dataset [98,99], a dataset used to detect the Parkinson's decease using voice measurements.
13. **Pima** dataset [100], a medical dataset.
14. **Popfailures** dataset [101], a dataset related to meterological data.
15. **Regions2** dataset, a medical dataset produced by liver biopsy images of patients with hepatitis C [102].
16. **Saheart** dataset [103], a medical dataset.
17. **Segment** dataset [104], a dataset related to image segmentation.
18. **Wdbc** dataset [105], a dataset used to detect breast tumors.
19. **Wine** dataset, a dataset related to chemical analysis of wines [106,107].
20. **Eeg** datasets [108,109], it is medical datasets about EEG signals and the following cases were used in the experiments:
    (a) Z_F_S,
    (b) ZO_NF_S
    (c) ZONF_S.
21. **Zoo** dataset [110].

The regression datasets used in the relevant experiments have as follows:

1. **Abalone** dataset [112], a dataset used to predict the age of abalones.
2. **Airfoil** dataset, a dataset provided by NASA [113] obtained from a series of aerodynamic and acoustic tests.
3. **Baseball** dataset, a dataset related to the salary of baseball players.
4. **BK** dataset [114], a dataset that was used to calculate the points in a basketball game.
5. **BL** dataset, used in machine problems.
6. **Concrete** dataset [115], a civil engineering dataset to calculate The concrete compressive strength

7. **Dee** dataset, used to estimate the daily average price of TkWhe electricity energy in Spain.
8. **Diabetes** dataset, a medical dataset.
9. **Housing** dataset [116].
10. **FA** dataset, used to fit body fat to other measurements.
11. **MB** dataset [117].
12. **MORTGAGE** dataset, holding economic data from USA. The goal is to predict the 30-Year Conventional Mortgage Rate.
13. **PY** dataset, (Pyrimidines problem)[118].
14. **Quake** dataset, used to approximate the strength of a earthquake given its the depth of its focal point, its latitude and its longitude.
15. **Treasure** dataset, which contains Economic data information of USA, where the the goal is to predict 1-Month CD Rate.

3.2. Experimental results

For greater reliability in the experimental results the method of 10 - fold cross validation was incorporated for every experimental dataset. Every experiment was repeated 30 times using different seed for the random generator each time. All the used code was implemented in ANSI C++ using the OPTIMUS programming library for optimization purposes, freely available from https://github.com/itsoulos/OPTIMUS/. For the classification datasets the average classification error as measured in the test set is reported while for the regression datasets the average regression error is reported. Also, in every table and additional column denoted as AVERAGE is added to show the average classification or regression error for the corresponding datasets. The values for the experimental parameters are shown in Table 2.

Table 2. The values for every parameter used in the experimens.

PARAMETER	MEANING	VALUE
m	Particles or Chromosomes	200
H	Number of hidden nodes	10
iter_{max}	Maximum number of iterations	200
L_{max}	Limit used in penalty calculation	100
λ	Penalty factor	100

The proposed technique that created artificial features is compared on the same datasets against a series of well - known method from the relevant literature:

1. A genetic algorithm with m chromosomes, denotes as GENETIC in the experimental tables. This genetic algorithm is used to train an artificial neural network with H hidden nodes. After the termination of the genetic algorithm the local optimization method BFGS is applied to the best chromosome of the population.
2. The Radial Basis Function (RBF) network [120] with H processing nodes.
3. The optimization method Adam [121], used to train an artificial neural netwok with H hidden nodes.
4. The Rprop optimization method [122–124], used to train an artificial neural network with H hidden nodes.
5. The NEAT method (NeuroEvolution of Augmenting Topologies) [125].

The experimental results using the above methods on the classification datasets are shown in Table 3 and the results for the regression datasets are illustrated in Table 4.

Table 3. Average classification error for the classification datasets using the well - known methods.

DATASET	GENETIC	RBF	ADAM	RPROP	NEAT
Appendicitis	18.10%	12.23%	16.50%	16.30%	17.20%
Australian	32.21%	34.89%	35.65%	36.12%	31.98%
Balance	8.97%	33.42%	7.87%	8.81%	23.14%
Bands	35.75%	37.22%	36.25%	36.32%	34.30%
Dermatology	30.58%	62.34%	26.14%	15.12%	32.43%
Hayes Roth	56.18%	64.36%	59.70%	37.46%	50.15%
Heart	28.34%	31.20%	38.53%	30.51%	39.27%
HouseVotes	6.62%	6.13%	7.48%	6.04%	10.89%
Ionosphere	15.14%	16.22%	16.64%	13.65%	19.67%
Liverdisorder	31.11%	30.84%	41.53%	40.26%	30.67%
Lymography	23.26%	25.31%	29.26%	24.67%	33.70%
Mammographic	19.88%	21.38%	46.25%	18.46%	22.85%
Parkinsons	18.05%	17.42%	24.06%	22.28%	18.56%
Pima	32.19%	25.78%	34.85%	34.27%	34.51%
Popfailures	5.94%	7.04%	5.18%	4.81%	7.05%
Regions2	29.39%	38.29%	29.85%	27.53%	33.23%
Saheart	34.86%	32.19%	34.04%	34.90%	34.51%
Segment	57.72%	59.68%	49.75%	52.14%	66.72%
Wdbc	8.56%	7.27%	35.35%	21.57%	12.88%
Wine	19.20%	31.41%	29.40%	30.73%	25.43%
Z_F_S	10.73%	13.16%	47.81%	29.28%	38.41%
ZO_NF_S	8.41%	9.02%	47.43%	6.43%	43.75%
ZONF_S	2.60%	4.03%	11.99%	27.27%	5.44%
ZOO	16.67%	21.93%	14.13%	15.47%	20.27%
AVERAGE	**22.94%**	**26.78%**	**30.24%**	**24.60%**	**28.63%**

Table 4. Average regression error using the well - known methods for the regression datasets.

DATASET	GENETIC	RBF	ADAM	RPROP	NEAT
ABALONE	7.17	7.37	4.30	4.55	9.88
AIRFOIL	0.003	0.27	0.005	0.002	0.067
BASEBALL	103.60	93.02	77.90	92.05	100.39
BK	0.027	0.02	0.03	1.599	0.15
BL	5.74	0.01	0.28	4.38	0.05
CONCRETE	0.0099	0.011	0.078	0.0086	0.081
DEE	1.013	0.17	0.63	0.608	1.512
DIABETES	19.86	0.49	3.03	1.11	4.25
HOUSING	43.26	57.68	80.20	74.38	56.49
FA	1.95	0.02	0.11	0.14	0.19
MB	3.39	2.16	0.06	0.055	0.061
MORTGAGE	2.41	1.45	9.24	9.19	14.11
PY	1.21	0.02	0.09	0.039	0.075
QUAKE	0.04	0.071	0.06	0.041	0.298
TREASURY	2.929	2.02	11.16	10.88	15.52
AVERAGE	**12.84**	**10.30**	**11.70**	**12.44**	**12.70**

The results using the proposed method and for the construction of 2,3 and 4 artificial features are presented in the relevant tables 5 and 6. The RBF column represents the experimental results in which after the construction of the artificial features, a RBF network with H processing nodes is applied on the modified dataset. Also, the column GENETIC in tables 5, 6 stands for the results obtained by the application of a genetic algorithm with

m chromosomes to the modified dataset, when the feature creation procedure has been finished.

Table 5. Experimental results for the classification datasets using the proposed method. Number in cells denote average classification error as measured on the test set.

	$f = 2$		$f = 3$		$f = 4$	
DATASET	RBF	GENETIC	RBF	GENETIC	RBF	GENETIC
APPENDICITIS	15.40%	14.33%	16.90%	15.77%	15.97%	17.30%
AUSTRALIAN	15.49%	14.48%	14.53%	15.33%	14.75%	15.97%
BALANCE	16.67%	2.89%	22.54%	4.94%	17.26%	4.62%
BANDS	38.09%	38.13%	37.09%	39.22%	37.22%	35.51%
DERMATOLOGY	41.58%	30.37%	35.46%	25.44%	40.45%	21.97%
HAYES ROTH	37.41%	27.92%	38.10%	25.74%	39.59%	25.82%
HEART	21.53%	17.13%	17.64%	16.87%	19.63%	15.69%
HOUSEVOTES	6.36%	3.78%	7.17%	3.25%	4.25%	3.52%
IONOSPHERE	10.32%	10.17%	10.12%	10.01%	11.42%	9.02%
LIVERDISORDER	34.23%	32.33%	35.84%	32.97%	35.93%	30.74%
LYMOGRAPHY	34.93%	28.67%	32.00%	23.00%	29.00%	23.83%
MAMMOGRAPHIC	16.92%	16.51%	16.47%	16.35%	17.54%	16.50%
PARKINSONS	11.14%	13.00%	11.30%	11.11%	12.95%	9.42%
PIMA	22.85%	22.76%	24.93%	24.67%	24.25%	24.20%
POPFAILURES	7.32%	7.41%	6.96%	7.62%	5.96%	5.67%
REGIONS2	28.52%	26.84%	24.91%	25.28%	25.35%	24.93%
SAHEART	29.29%	28.63%	28.92%	30.31%	28.25%	30.25%
SEGMENT	52.69%	45.59%	46.83%	41.06%	50.15%	39.52%
WDBC	5.00%	4.66%	5.76%	4.97%	5.13%	3.84%
WINE	8.92%	7.22%	6.76%	5.75%	6.00%	5.86%
Z_F_S	7.91%	8.37%	7.89%	7.67%	5.21%	6.86%
ZO_NF_S	6.90%	6.85%	6.95%	5.65%	6.24%	5.28%
ZONF_S	3.08%	3.40%	2.44%	2.52%	3.47%	3.33%
ZOO	26.47%	7.83%	31.73%	10.03%	28.70%	11.57%
AVERAGE	**20.79%**	**17.47%**	**20.39%**	**16.90%**	**20.19%**	**16.30%**

Table 6. Experimental results on the regression datasets using the proposed method. The number in cells denote average regression error as measured on the test set.

	$f = 2$		$f = 3$		$f = 4$	
DATASET	RBF	GENETIC	RBF	GENETIC	RBF	GENETIC
ABALONE	4.361	3.518	4.159	3.839	4.859	3.786
AIRFOIL	0.003	0.001	0.003	0.001	0.003	0.001
BASEBALL	66.00	53.74	60.79	57.04	66.19	61.69
BK	0.022	0.031	0.021	0.029	0.019	0.023
BL	0.413	0.0001	0.019	0.007	0.043	0.011
CONCRETE	0.008	0.006	0.007	0.005	0.008	0.004
DEE	0.259	0.252	0.339	0.286	0.609	0.5
DIABETES	0.611	0.832	0.634	1.411	0.857	1.157
HOUSING	22.387	15.583	18.614	13.602	14.83	13.208
FA	0.056	0.011	0.015	0.011	0.015	0.012
MB	0.258	0.087	0.115	0.078	0.342	0.072
MORTGAGE	0.621	0.046	0.65	0.037	0.078	0.04
PY	2.894	0.14	0.936	0.029	0.724	0.031
QUAKE	0.069	0.036	0.057	0.037	0.04	0.037
TREASURY	0.912	0.088	0.874	0.084	0.173	0.076
AVERAGE	**6.59**	**4.96**	**5.82**	**5.10**	**5.92**	**5.38**

As can be seen from the experimental results, the proposed technique is able to significantly reduce the error in the corresponding test sets. Especially in the case of regression problems, the reduction in error is on average greater than 50%. Moreover, the usage of a neural network trained by a genetic algorithm on the modified datasets, gives clearly better results than the use of an RBF neural network, especially in the classification datasets.

(Na grafei pos ta features exon ftiaxtei me RBF epeidi einai grigori diadikasia)
(Na grafei pos yparxei simantiki meiosi stin diastasi ton dedomenon eisodou)
(Grafimata apo alexandros)

4. Conclusions

A hybrid technique that utilizes a Particle Swarm Optimizer and a feature creation method using Grammatical Evolution was introduced here. The proposed method can identify possible dependecies between the original features and also can reduce the number of required features to a limited number. Also, the method can remove from the set of features those features that may not contribute to the learning of the data set by some machine learning model. In addition, to make learning more efficient the values of the generated features are bounded within a value interval using penalty factors. The constructed features are evaluated in terms of their effectiveness with the help of a fast machine learning model such as the RBF network, even though other more effective models could also be used. Among the advantages of the proposed procedure is the fact that it does not require any prior knowledge of the data set to which it will be applied and furthermore the procedure is exactly the same whether it is a data classification problem or a data fitting problem. The Particle Swarm Optimization method was used for the production of the characteristics as it has been proven by the relevant literature to be an extremely efficient technique and has a limited number of parameters that must be defined by the user.

The proposed method was applied on a extended series of widely used datasets from the relevant literature and was compared against some machine learning models on the same datasets. From the experimental results, it was seen that the proposed technique dramatically improves the performance of traditional learning techniques when applied to artificial features. This improvement reaches an average of 30% for data classification and 50% for data fitting problems. Furthermore, as shown in the experimental results, the proposed technique is able to give excellent results even when only two features are used. Future extensions of the method may include the use of parallel techniques for feature construction to drastically reduce the required execution time.

Author Contributions: I.G.T. and A.T. conceived of the idea and the methodology and I.G.T has implemented the corresponding software. I.G.T. conducted the experiments, employing objective functions as test cases, and provided the comparative experiments. A.T. has performed the necessary statistical tests. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Institutional Review Board Statement: Not applicable.

Acknowledgments: The experiments of this research work were performed at the high performance computing system established at Knowledge and Intelligent Computing Laboratory, Department of Informatics and Telecommunications, University of Ioannina, acquired with the project "Educational Laboratory equipment of TEI of Epirus" with MIS 5007094 funded by the Operational Programme "Epirus" 2014–2020, by ERDF and national funds.

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

Sample Availability: Not applicable.

References

1. E.M. Metodiev, B. Nachman, J. Thaler, Classification without labels: learning from mixed samples in high energy physics. J. High Energ. Phys. 2017, article number 174, 2017.
2. P. Baldi, K. Cranmer, T. Faucett et al, Parameterized neural networks for high-energy physics, Eur. Phys. J. C **76**, 2016.
3. J. J. Valdas and G. Bonham-Carter, Time dependent neural network models for detecting changes of state in complex processes: Applications in earth sciences and astronomy, Neural Networks **19**, pp. 196-207, 2006
4. G. Carleo,M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science **355**, pp. 602-606, 2017.
5. C. Güler, G. D. Thyne, J. E. McCray, K.A. Turner, Evaluation of graphical and multivariate statistical methods for classification of water chemistry data, Hydrogeology Journal **10**, pp. 455-474, 2002
6. E. Byvatov ,U. Fechner ,J. Sadowski , G. Schneider, Comparison of Support Vector Machine and Artificial Neural Network Systems for Drug/Nondrug Classification, J. Chem. Inf. Comput. Sci. **43**, pp 1882–1889, 2003.
7. Kunwar P. Singh, Ankita Basant, Amrita Malik, Gunja Jain, Artificial neural network modeling of the river water quality—A case study, Ecological Modelling **220**, pp. 888-895, 2009.
8. I. Kaastra, M. Boyd, Designing a neural network for forecasting financial and economic time series, Neurocomputing **10**, pp. 215-236, 1996.
9. Moshe Leshno, Yishay Spector, Neural network prediction analysis: The bankruptcy case, Neurocomputing **10**, pp. 125-147, 1996.
10. A. Astel, S. Tsakovski, V, Simeonov et al., Multivariate classification and modeling in surface water pollution estimation. Anal Bioanal Chem **390**, pp. 1283–1292, 2008.
11. A. Azid, H. Juahir, M.E. Toriman et al., Prediction of the Level of Air Pollution Using Principal Component Analysis and Artificial Neural Network Techniques: a Case Study in Malaysia, Water Air Soil Pollut **225**, pp. 2063, 2014.
12. H. Maleki, A. Sorooshian, G. Goudarzi et al., Air pollution prediction by using an artificial neural network model, Clean Techn Environ Policy **21**, pp. 1341–1352, 2019.
13. Igor I. Baskin, David Winkler and Igor V. Tetko, A renaissance of neural networks in drug discovery, Expert Opinion on Drug Discovery **11**, pp. 785-795, 2016.
14. Ronadl Bartzatt, Prediction of Novel Anti-Ebola Virus Compounds Utilizing Artificial Neural Network (ANN), Chemistry Faculty Publications **49**, pp. 16-34, 2018.
15. Y. Wu, K. Ianakiev, V. Govindaraju, Improved k-nearest neighbor classification, Pattern Recognition **35**, pp. 2311-2318, 2002.
16. Z.-ga Liu, Q. Pan, J. Dezert, A new belief-based K-nearest neighbor classification method, Pattern Recognition **46**, pp. 834-844, 2012.
17. C. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
18. G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of Control Signals and Systems **2**, pp. 303-314, 1989.
19. H. Yu, T. Xie, S. Paszczynski, B. M. Wilamowski, Advantages of Radial Basis Function Networks for Dynamic System Design, in IEEE Transactions on Industrial Electronics **58**, pp. 5438-5450, 2011.
20. D. Chen, Research on Traffic Flow Prediction in the Big Data Environment Based on the Improved RBF Neural Network, IEEE Transactions on Industrial Informatics **13**, pp. 2000-2008, 2017.
21. Z. Yang, M. Mourshed, K. Liu, X. Xu, S. Feng, A novel competitive swarm optimized RBF neural network model for short-term solar power generation forecasting, Neurocomputing **397**, pp. 415-421, 2020.
22. I. Steinwart, A. Christmann, Support Vector Machines, Information Science and Statistics, Springer, 2008.
23. A. Iranmehr, H. Masnadi-Shirazi, N. Vasconcelos, Cost-sensitive support vector machines, Neurocomputing **343**, pp. 50-64, 2019.
24. S.B. Kotsiantis, Decision trees: a recent overview, Artif Intell Rev **39**, pp. 261–283, 2013.
25. D. Bertsimas, J. Dunn, Optimal classification trees, Mach Learn **106**, pp. 1039–1082, 2017.
26. S.B. Kotsiantis, I.D. Zaharakis, P.E. Pintelas, Machine learning: a review of classification and combining techniques, Artif Intell Rev **26**, pp. 159–190, 2006.
27. D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning representations by back-propagating errors, Nature **323**, pp. 533 - 536 , 1986.
28. T. Chen and S. Zhong, Privacy-Preserving Backpropagation Neural Network Learning, IEEE Transactions on Neural Networks **20**, , pp. 1554-1564, 2009.
29. F. H. F. Leung, H. K. Lam, S. H. Ling and P. K. S. Tam, Tuning of the structure and parameters of a neural network using an improved genetic algorithm, IEEE Transactions on Neural Networks **14**, pp. 79-88, 2003.
30. A. Sedki, D. Ouazar, E. El Mazoudi, Evolving neural network using real coded genetic algorithm for daily rainfall–runoff forecasting, Expert Systems with Applications **36**, pp. 4523-4527, 2009.
31. A. Majdi, M. Beiki, Evolving neural network using a genetic algorithm for predicting the deformation modulus of rock masses, International Journal of Rock Mechanics and Mining Sciences **47**, pp. 246-253, 2010.
32. M.J.D Powell, A Tolerant Algorithm for Linearly Constrained Optimization Calculations, Mathematical Programming **45**, pp. 547-566, 1989.
33. M. Verleysen, D. Francois, G. Simon, V. Wertz, On the effects of dimensionality on data analysis with neural networks. In: Mira J., Álvarez J.R. (eds) Artificial Neural Nets Problem Solving Methods. IWANN 2003. Lecture Notes in Computer Science, vol 2687. Springer, Berlin, Heidelberg. 2003.

34. 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.
35. 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.
36. 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.
37. Hanchuan Peng, Fuhui Long, and Chris Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence **27**, pp.1226-1238, 2005.
38. Chris Ding, and Hanchuan Peng, Minimum redundancy feature selection from microarray gene expression data, Journal of Bioinformatics and Computational Biology **3**, pp.185-205, 2005.
39. Y. Wang, H. Yao, S. Zhao, Auto-encoder based dimensionality reduction, Neurocomputing **184**, pp. 232-242, 2016.
40. S. Geman, E. Bienenstock and R. Doursat, Neural networks and the bias/variance dilemma, Neural Computation 4 , pp. 1 - 58, 1992.
41. Douglas M. Hawkins, The Problem of Overfitting, J. Chem. Inf. Comput. Sci. **44**, pp. 1–12, 2004.
42. S.J. Nowlan and G.E. Hinton, Simplifying neural networks by soft weight sharing, Neural Computation 4, pp. 473-493, 1992.
43. W. Roth, F. Pernkopf, Bayesian Neural Networks with Weight Sharing Using Dirichlet Processes, IEEE Transactions on Pattern Analysis and Machine Intelligence **42**, pp. 246-252, 2020.
44. S.J. Hanson and L.Y. Pratt, Comparing biases for minimal network construction with back propagation, In D.S. Touretzky (Ed.), Advances in Neural Information Processing Systems, Volume 1, pp. 177-185, San Mateo, CA: Morgan Kaufmann, 1989.
45. M.C. Mozer and P. Smolensky, Skeletonization: a technique for trimming the fat from a network via relevance assesment. In D.S. Touretzky (Ed.), Advances in Neural Processing Systems, Volume 1, pp. 107-115, San Mateo CA: Morgan Kaufmann, 1989.
46. M. Augasta and T. Kathirvalavakumar, Pruning algorithms of neural networks — a comparative study, Central European Journal of Computer Science, 2003.
47. F. Hergert, W. Finnoff, H. G. Zimmermann, A comparison of weight elimination methods for reducing complexity in neural networks, In: Proceedings 1992] IJCNN International Joint Conference on Neural Networks, Baltimore, MD, USA, pp. 980-987 vol.3, 1992.
48. M. Cottrell, B. Girard, Y. Girard, M. Mangeas, C. Muller, Neural modeling for time series: A statistical stepwise method for weight elimination, IEEE Transactions on Neural Networks **6**, pp. 1355-1364, 1995.
49. C. M. Ennett, M. Frize, Weight-elimination neural networks applied to coronary surgery mortality prediction, IEEE Transactions on Information Technology in Biomedicine **7**, pp. 86-92, 2003.
50. N. K. Treadgold and T. D. Gedeon, Simulated annealing and weight decay in adaptive learning: the SARPROP algorithm,IEEE Transactions on Neural Networks **9**, pp. 662-668, 1998.
51. M. Carvalho and T. B. Ludermir, Particle Swarm Optimization of Feed-Forward Neural Networks with Weight Decay, 2006 Sixth International Conference on Hybrid Intelligent Systems (HIS'06), Rio de Janeiro, Brazil, 2006, pp. 5-5.
52. M. O'Neill, C. Ryan, Grammatical evolution, IEEE Trans. Evol. Comput. **5,**pp. 349–358, 2001.
53. Alfonso Ortega, Rafael Sánchez, Manuel Alfonseca Moreno, Automatic composition of music by means of grammatical evolution, APL '02 Proceedings of the 2002 conference on APL: array processing languages: lore, problems, and applications Pages 148 - 155.
54. Michael O'Neill, Anthony Brabazon, Conor Ryan, J. J. Collins, Evolving Market Index Trading Rules Using Grammatical Evolution, Applications of Evolutionary Computing Volume 2037 of the series Lecture Notes in Computer Science pp 343-352.
55. M. O'Neill, C. Ryan, Grammatical Evolution: Evolutionary Automatic Programming in a Arbitary Language, Genetic Programming, vol. 4, Kluwer Academic Publishers, Dordrecht, 2003
56. J.J. Collins, C. Ryan, in: Proceedings of AROB 2000, Fifth International Symposium on Artificial Life and Robotics, 2000.
57. M. O'Neill, C. Ryan, in: K. Miettinen, M.M. Mkel, P. Neittaanmki, J. Periaux (Eds.), Evolutionary Algorithms in Engineering and Computer Science, Jyvskyl, Finland, 1999, pp. 127–134.
58. J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.
59. Riccardo Poli, James Kennedy kennedy, Tim Blackwell, Particle swarm optimization An Overview, Swarm Intelligence **1**, pp 33-57, 2007.
60. Ioan Cristian Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Information Processing Letters **85**, pp. 317-325, 2003.
61. Anderson Alvarenga de Moura Meneses, Marcelo Dornellas, Machado Roberto Schirru, Particle Swarm Optimization applied to the nuclear reload problem of a Pressurized Water Reactor, Progress in Nuclear Energy **51**, pp. 319-326, 2009.
62. Ranjit Shaw, Shalivahan Srivastava, Particle swarm optimization: A new tool to invert geophysical data, Geophysics **72**, 2007.
63. C. O. Ourique, E.C. Biscaia, J.C. Pinto, The use of particle swarm optimization for dynamical analysis in chemical processes, Computers & Chemical Engineering **26**, pp. 1783-1793, 2002.
64. H. Fang, J. Zhou, Z. Wang et al, Hybrid method integrating machine learning and particle swarm optimization for smart chemical process operations, Front. Chem. Sci. Eng. **16**, pp. 274–287, 2022.

65. M.P. Wachowiak, R. Smolikova, Yufeng Zheng, J.M. Zurada, A.S. Elmaghraby, An approach to multimodal biomedical image registration utilizing particle swarm optimization, IEEE Transactions on Evolutionary Computation **8**, pp. 289-301, 2004.
66. Yannis Marinakis. Magdalene Marinaki, Georgios Dounias, Particle swarm optimization for pap-smear diagnosis, Expert Systems with Applications **35**, pp. 1645-1656, 2008.
67. Jong-Bae Park, Yun-Won Jeong, Joong-Rin Shin, Kwang Y. Lee, An Improved Particle Swarm Optimization for Nonconvex Economic Dispatch Problems, IEEE Transactions on Power Systems **25**, pp. 156-16**216**6, 2010.
68. B. Liu, L. Wang, Y.H. Jin, An Effective PSO-Based Memetic Algorithm for Flow Shop Scheduling, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **37**, pp. 18-27, 2007.
69. J. Yang, L. He, S. Fu, An improved PSO-based charging strategy of electric vehicles in electrical distribution grid, Applied Energy **128**, pp. 82-92, 2014.
70. K. Mistry, L. Zhang, S. C. Neoh, C. P. Lim, B. Fielding, A Micro-GA Embedded PSO Feature Selection Approach to Intelligent Facial Emotion Recognition, IEEE Transactions on Cybernetics. **47**, pp. 1496-1509, 2017.
71. S. Han, X. Shan, J. Fu, W. Xu, H. Mi, Industrial robot trajectory planning based on improved pso algorithm, J. Phys.: Conf. Ser. **1820**, 012185, 2021.
72. Dimitris Gavrilis, Ioannis G. Tsoulos, Evangelos Dermatas, Selecting and constructing features using grammatical evolution, Pattern Recognition Letters **29**,pp. 1358-1365, 2008.
73. 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.
74. 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
75. 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.
76. I.G. Tsoulos, C. Stylios, V. Charalampous, COVID-19 Predictive Models Based on Grammatical Evolution, SN COMPUT. SCI. **4**, 191, 2023.
77. V. Christou, I.G. Tsoulos, V. Loupas, A.T. Tzallas, C. Gogos, P.S. Karvelis, N. Antoniadis, E. Glavas, N. Giannakeas, Performance and early drop prediction for higher education students using machine learning, Expert Systems with Applications **225**, 120079, 2023.
78. A. Verikas, M. Bacauskiene, Feature selection with neural networks, Pattern Recognition Letters **23**, pp. 1323-1335, 2002.
79. Md. Monirul Kabir, Md. Monirul Islam, K. Murase, A new wrapper feature selection approach using neural network, Neurocomputing **73**, pp. 3273-3283, 2010.
80. V.S. Devi, Class Specific Feature Selection Using Simulated Annealing. In: Prasath, R., Vuppala, A., Kathirvalavakumar, T. (eds) Mining Intelligence and Knowledge Exploration. MIKE 2015. Lecture Notes in Computer Science(), vol 9468. Springer, Cham. 2015.
81. K. Neshatian, M. Zhang, P. Andreae, A Filter Approach to Multiple Feature Construction for Symbolic Learning Classifiers Using Genetic Programming, IEEE Transactions on Evolutionary Computation **16**, pp. 645-661, 2012.
82. 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.
83. J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing 17, pp. 255-287, 2011.
84. Weiss, Sholom M. and Kulikowski, Casimir A., Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems, Morgan Kaufmann Publishers Inc, 1991.
85. M. Wang, Y.Y. Zhang, F. Min, Active learning through multi-standard optimization, IEEE Access **7**, pp. 56772–56784, 2019.
86. J.R. Quinlan, Simplifying Decision Trees. International Journal of Man-Machine Studies **27**, pp. 221-234, 1987.
87. T. Shultz, D. Mareschal, W. Schmidt, Modeling Cognitive Development on Balance Scale Phenomena, Machine Learning **16**, pp. 59-88, 1994.
88. B. Evans, D. Fisher, Overcoming process delays with decision tree induction. IEEE Expert **9**, pp. 60-66, 1994.
89. G. Demiroz, H.A. Govenir, N. Ilter, Learning Differential Diagnosis of Eryhemato-Squamous Diseases using Voting Feature Intervals, Artificial Intelligence in Medicine. **13**, pp. 147–165, 1998.
90. I. Kononenko, E. Šimec, M. Robnik-Šikonja, Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF, Applied Intelligence **7**, pp. 39–55, 1997
91. B. Hayes-Roth, B., F. Hayes-Roth. Concept learning and the recognition and classification of exemplars. Journal of Verbal Learning and Verbal Behavior **16**, pp. 321-338, 1977.
92. R.M. French, N. Chater, Using noise to compute error surfaces in connectionist networks: a novel means of reducing catastrophic forgetting, Neural Comput. **14**, pp. 1755-1769, 2002.
93. J.G. Dy , C.E. Brodley, Feature Selection for Unsupervised Learning, The Journal of Machine Learning Research **5**, pp 845–889, 2004.

94. S. J. Perantonis, V. Virvilis, Input Feature Extraction for Multilayered Perceptrons Using Supervised Principal Component Analysis, Neural Processing Letters **10**, pp 243–252, 1999.
95. J. Mcdermott, R.S. Forsyth, Diagnosing a disorder in a classification benchmark, Pattern Recognition Letters **73**, pp. 41-43, 2016.
96. J. Garcke, M. Griebel, Classification with sparse grids using simplicial basis functions, Intell. Data Anal. **6**, pp. 483-502, 2002.
97. M. Elter, R. Schulz-Wendtland, T. Wittenberg, The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process, Med Phys. **34**, pp. 4164-72, 2007.
98. M.A. Little, P.E. McSharry, S.J Roberts et al, Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection. BioMed Eng OnLine **6**, 23, 2007.
99. M.A. Little, P.E. McSharry, E.J. Hunter, J. Spielman, L.O. Ramig, Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. IEEE Trans Biomed Eng. **56**, pp. 1015-1022, 2009.
100. J.W. Smith, J.E. Everhart, W.C. Dickson, W.C. Knowler, R.S. Johannes, Using the ADAP learning algorithm to forecast the onset of diabetes mellitus, In: Proceedings of the Symposium on Computer Applications and Medical Care IEEE Computer Society Press, pp.261-265, 1988.
101. D.D. Lucas, R. Klein, J. Tannahill, D. Ivanova, S. Brandon, D. Domyancic, Y. Zhang, Failure analysis of parameter-induced simulation crashes in climate models, Geoscientific Model Development **6**, pp. 1157-1171, 2013.
102. N. Giannakeas, M.G. Tsipouras, A.T. Tzallas, K. Kyriakidi, Z.E. Tsianou, P. Manousou, A. Hall, E.C. Karvounis, V. Tsianos, E. Tsianos, A clustering based method for collagen proportional area extraction in liver biopsy images (2015) Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, 2015-November, art. no. 7319047, pp. 3097-3100.
103. T. Hastie, R. Tibshirani, Non-parametric logistic and proportional odds regression, JRSS-C (Applied Statistics) **36**, pp. 260–276, 1987.
104. M. Dash, H. Liu, P. Scheuermann, K. L. Tan, Fast hierarchical clustering and its validation, Data & Knowledge Engineering **44**, pp 109–138, 2003.
105. W.H. Wolberg, O.L. Mangasarian, Multisurface method of pattern separation for medical diagnosis applied to breast cytology, Proc Natl Acad Sci U S A. **87**, pp. 9193–9196, 1990.
106. M. Raymer, T.E. Doom, L.A. Kuhn, W.F. Punch, Knowledge discovery in medical and biological datasets using a hybrid Bayes classifier/evolutionary algorithm. IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society, **33** , pp. 802-813, 2003.
107. P. Zhong, M. Fukushima, Regularized nonsmooth Newton method for multi-class support vector machines, Optimization Methods and Software **22**, pp. 225-236, 2007.
108. R. G. Andrzejak, K. Lehnertz, F.Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state," Physical Review E, vol. 64, no. 6, Article ID 061907, 8 pages, 2001.
109. A. T. Tzallas, M. G. Tsipouras, and D. I. Fotiadis, "Automatic Seizure Detection Based on Time-Frequency Analysis and Artificial Neural Networks," Computational Intelligence and Neuroscience, vol. 2007, Article ID 80510, 13 pages, 2007. doi:10.1155/2007/80510
110. M. Koivisto, K. Sood, Exact Bayesian Structure Discovery in Bayesian Networks, The Journal of Machine Learning Research **5**, pp. 549–573, 2004.
111. R.G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state, Phys. Rev. E **64**, pp. 1-8, 2001.
112. W. J Nash, T.L. Sellers, S.R. Talbot, A.J. Cawthor, W.B. Ford, The Population Biology of Abalone (_Haliotis_ species) in Tasmania. I. Blacklip Abalone (_H. rubra_) from the North Coast and Islands of Bass Strait, Sea Fisheries Division, Technical Report No. 48 (ISSN 1034-3288), 1994.
113. T.F. Brooks, D.S. Pope, A.M. Marcolini, Airfoil self-noise and prediction. Technical report, NASA RP-1218, July 1989.
114. J.S. Simonoff, Smooting Methods in Statistics, Springer - Verlag, 1996.
115. I.Cheng Yeh, Modeling of strength of high performance concrete using artificial neural networks, Cement and Concrete Research. **28**, pp. 1797-1808, 1998.
116. D. Harrison and D.L. Rubinfeld, Hedonic prices and the demand for clean ai, J. Environ. Economics & Management **5**, pp. 81-102, 1978.
117. J.S. Simonoff, Smooting Methods in Statistics, Springer - Verlag, 1996.
118. R.D. King, S. Muggleton, R. Lewis, M.J.E. Sternberg, Proc. Nat. Acad. Sci. USA **89**, pp. 11322–11326, 1992.
119. R. Fletcher, A new approach to variable metric algorithms, Computer Journal **13**, pp. 317-322, 1970.
120. J. Park and I. W. Sandberg, Universal Approximation Using Radial-Basis-Function Networks, Neural Computation **3**, pp. 246-257, 1991.
121. D. P. Kingma, J. L. Ba, ADAM: a method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), pp. 1–15, 2015.
122. M. Riedmiller and H. Braun, A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP algorithm, Proc. of the IEEE Intl. Conf. on Neural Networks, San Francisco, CA, pp. 586–591, 1993.

123. T. Pajchrowski, K. Zawirski and K. Nowopolski, Neural Speed Controller Trained Online by Means of Modified RPROP Algorithm, IEEE Transactions on Industrial Informatics **11**, pp. 560-568, 2015.
124. Rinda Parama Satya Hermanto, Suharjito, Diana, Ariadi Nugroho, Waiting-Time Estimation in Bank Customer Queues using RPROP Neural Networks, Procedia Computer Science **135**, pp. 35-42, 2018.
125. K. O. Stanley, R. Miikkulainen, Evolving Neural Networks through Augmenting Topologies, Evolutionary Computation **10**, pp. 99-127, 2002.