MDPI

Article

Improving the generalization abilities of constructed neural networks with the addition of local optimization techniques

Ioannis G. Tsoulos[1,*], Vasileios Charilogis[2], Dimitrios Tsalikakis[3], Alexandros Tzallas[4]

[1] Department of Informatics and Telecommunications, University of Ioannina, Greece;itsoulos@uoi.gr
[2] Department of Informatics and Telecommunications, University of Ioannina, Greece; v.charilog@uoi.gr
[3] Department of Engineering Informatics and Telecommunications, University of Western Macedonia, 50100 Kozani, Greece;tsalikakis@gmail.com
[4] Department of Informatics and Telecommunications, University of Ioannina, Greece; tzallas@uoi.gr
* Correspondence: itsoulos@uoi.gr

Abstract: Constructed neural networks with the assistance of Grammatical Evolution have been widely used in a series of classification and regression problems in the recent literature of machine learning. Application areas of this innovative machine learning technique include solving differential equations, autism screening, measuring motor function in Parkinson's disease. Although this technique has given excellent results, in many cases it is trapped in local minimum and cannot perform satisfactorily in many problems. For this purpose, it is considered necessary to find techniques to avoid local minima and one technique is the periodic application of local minimization techniques that will undertake to adjust the parameters of the constructed artificial neural network, but maintaining the already existing architecture created by Grammatical Evolution. Periodic application of local minimization techniques has shown a significant reduction in both classification and data fitting problems found in the relevant literature.

Keywords: Grammatical Evolution; Genetic Programming; Neural networks; Local Optimization

Citation: Tsoulos, I.G.; Charilogis, V.; Tsalikakis, D.; Tzallas A. Improving the generalization abilities of constructed neural networks with the addition of local optimization techniques. *Journal Not Specified* **2024**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

1. Introduction

Artificial neural networks (ANNs) are parametric machine learning models [1,2], in which a set of parameters, commonly called weights, must be estimated in order for this model to adapt to classification or regression data. Neural networks have been used in a variety of scientific and real - world problems, such as physics problems [3–5], solutions of differential equations [6,7], solar radiation prediction [8], agriculture problems [9,10], problems derived from chemistry [11–13], wind speed forecasting [14], problems that appear in economic sciences [15–17], problems related to medicine [18,19] etc. A neural network can be expressed as a function $N(\overrightarrow{x}, \overrightarrow{w})$, where the vector $\overrightarrow{x}$ represents the input vector to the neural network and the vector $\overrightarrow{w}$ stands for the vector of parameters that should be calculated. The set of parameters is calculated by minimizing the so-called training error, which is defined as:

$$E\left(N\left(\overrightarrow{x}, \overrightarrow{w}\right)\right) = \sum_{i=1}^{M}\left(N\left(\overrightarrow{x}_i, \overrightarrow{w}\right) - y_i\right)^2 \tag{1}$$

In equation 1 the set $\left(\overrightarrow{x_i}, y_i\right),\ i = 1, ..., M$ stands for the train set of the neural network. The values y_i denote the target outputs for patterns $\overrightarrow{x_i}$. In the recent bibliography various methods have appeared that minimize this equation, such as the Back Propagation method [20,21], the RPROP method [22,23], the ADAM method [24] etc. Furthermore, global optimization techniques were also used to train neural networks, such as the Simulated Annealing method [25,26], genetic Algorithms [27,28], Particle Swarm Optimization (PSO) [29], Differential Evolution [30], Ant Colony Optimization [31], Gray Wolf Optimizer [32], Whale optimization [33] etc.

In many cases, especially when the data is large in volume or has a high number of features, significant times are observed in the training of artificial neural networks by optimization methods. For this reason, techniques have been presented in recent years that exploit modern parallel computing structures for faster training of these machine learning models [34–36].

Another important aspect in artificial neural networks is the initialization of parameters. Various techniques have been proposed in this area, such as usage of polynomial bases [37], initialization based on decision trees [38], usage of intervals [39], discriminant learning [40] etc. Also, recently Chen et al. proposed a new weight initialization method that is based on the linear product structure for neural networks [41].

Identifying the optimal architecture of an artificial neural network plays an important role in the generalization ability of a network. Networks with a few number of neurons can be trained faster and may have good generalization abilities, but in many cases the optimization method gets trapped in local minima of the error function. On the other hand, networks with many neurons can have a significantly reduced training error but require a large computational time for their training and many times do not perform significantly when applied to data that is not present in the training set. In this direction, many researchers proposed various methods to discover the optimal architecture, such as incorporation of genetic algorithms [42,43], application of the Particle Swarm Optimization method [44], usage of reinforcement learning [45] etc. Also, Islam et al. proposed a new adaptive merging and growing algorithm for the design of neural networks [46].

Recently, a technique was presented that utilizes the Grammatical Evolution method [47] for the efficient construction of the architecture of an artificial neural network as well as the calculation of the optimal values of the parameters [48]. In this technique, the architecture of the neural network is found and, at the same time, those features are selected which will reduce the training error. In this way, the number of required features can be drastically reduced, leading to neural networks that are faster in response and with better generalization abilities. The neural construction technique has been applied in a variety of cases, such as location of amide I bonds [49], solving differential equations [50], application in data collected for Parkinson's disease [51], performance and early drop prediction for higher education students [52], autism screening [53] etc. Also, a software that implements this method can be downloaded freely from https://github.com/itsoulos/NNC (accessed on 11 September 2024) [54].

Although the neural network construction technique has been successfully used in a variety of applications and is able to construct the structure of a neural network as well as find satisfactory values for the model parameters, it can often get trapped in local minima of the error function, which results in reduced performance in the problems to be solved. In this research paper, the periodic application of local optimization techniques is proposed in randomly selected artificial neural networks constructed by Grammatical Evolution. Local optimization does not alter the generated neural network structure but can more efficiently identify values of the network parameters with lower values of the training error. The proposed method was tested on a series of classification and regression datasets from the relevant literature, and it seems to reduce the test error obtained by the original neural construction technique.

The rest of this article is organized as follows: in section 2 the main aspects of the proposed method are discussed in detail, in section 3 the conducted experiments are shown and finally in section 4 some conclusions and guidelines for future research are discussed.

2. Method description

This section initiates with a description of the Grammatical Evolution procedure and continues with the detail description of the proposed algorithm.

2.1. Grammatical evolution

Grammatical evolution is a genetic algorithm, where the chromosomes stand for production rules of a provided BNF (Backus–Naur form) grammar [55]. The method was used on a series of real world cases, such as approximation of functions from experimental data [56,57], application in trigonometric problems [58], automatic composition of music [59], production of numeric constants with an arbitrary number of digits [60], **vi**deo games [61,62], energy problems [63], combinatorial optimization [64], cryptography [65], production of decision trees [66], electronics [67], Wikipedia taxonomies [68], economics [69], bioinformatics [70] etc. A BNF grammar is commonly defined as a set $G = (N, T, S, P)$ where

- The set N contains the non-terminal symbols of the grammar.
- The set T contains the terminal symbols of the grammar.
- The non - terminal symbol S represents the start symbol of the grammar.
- The set P contains the production rules of the grammar. These rule are used to produce terminal symbols from non - terminal symbols, and they are in the form $A \to a$ or $A \to aB,\ A, B \in N,\ a \in T$.

The production algorithm starts from the symbol S and through a series of steps, produces valid programs by replacing non-terminal symbols with the right hand of the selected production rule. The selection of the production rules is done in two steps:

- Obtain the next element V from the chromosome which is processed.
- The production rule is selected according to: Rule = V mod N_R, where N_R is the total number of production rules for the current non – terminal symbol.

The grammar used in the neural construction method is shown in Figure 1. The number in parentheses denote the sequence number of each rule for every non - terminal symbol. The constant n defines the number of features for the used dataset.

```
S:=<sigexpr>                              (0)
<sigexpr>::=<Node>                        (0)
          | <Node> + <sigexpr>            (1)
<Node>::=<number>*sig(<sum>+<number>) (0)
<sum>::= <number>*<xxlist>                (0)
          |    <sum>+<sum>                (1)
<xxlist>::= x1        (0)
             |    x2  (1)
             ..............
             |    xn  (n-1)
<number>::= (<digitlist>.<digitlist>)        (0)
             |    (-<digitlist>.<digitlist>) (1)
<digitlist>::= <digit>            (0)
             | <digit><digitlist> (1)
<digit>::= 0      (0)
             |  1 (1)
             ...........
             |  9 (9)
```

Figure 1. The grammar used by the neural construction method.

This grammar can produce artificial neural networks in the form:

$$N(\vec{x}, \vec{w}) = \sum_{i=1}^{H} w_{(n+2)i-(n+1)} \sigma \left(\sum_{j=1}^{n} x_j w_{(n+2)i-(n+1)+j} + w_{(n+2)i} \right) \tag{2}$$

where the parameter H defines the number of processing units. The function $\sigma(x)$ is commonly called sigmoid function, and it has the following definition:

$$\sigma(x) = \frac{1}{1+\exp(-x)} \tag{3}$$

2.2. The proposed algorithm

The main steps of the current method are derived from the steps of the original neural network construction method with the addition of the periodical application of the local optimization algorithm. The steps of the method are given below:

1. **Initialization step**.
   (a) **Set** $k = 0$ the generation counter.
   (b) **Set** as N_g the maximum number of generations and as N_c the number of chromosomes in the genetic population.
   (c) **Set** as p_s the selection rate **and** as p_m the mutation rate of the genetic algorithm.
   (d) **Set** as N_T the number of chromosomes the will be selected to apply the local search optimization method on them.
   (e) **Set** as N_I the number of generations that must pass before applying the local optimization technique.
   (f) **Set** as F the range of values within which the local optimization method can vary the parameters of the neural network.
   (g) **Initialize** the chromosomes g_i, $i = 1, \ldots, N_c$ as sets of random integers.
2. **Fitness Calculation step**.
   (a) **For** $i = 1, \ldots, N_c$ **do**
      i. **Create** the neural network $N_i(\overrightarrow{x}, \overrightarrow{w_i})$ for the chromosome g_i using the procedure of subsection 2.1. The vector $\overrightarrow{w_i}$ denotes the set of parameters produced for the chromosome g_i.
      ii. **Set** as $f_i = \sum_{j=1}^{M} \left(N_i(\overrightarrow{x}_j, \overrightarrow{w_i}) - y_j\right)^2$ the fitness of chromosome i. The set $(\overrightarrow{x_j}, y_j)$, $j = 1, ..., M$ represents the train set.
   (b) **EndFor**
3. **Genetic operations step**.
   (a) **Apply** the selection procedure: Initially the chromosomes are sorted according to their fitness. The $(1 - p_s) \times N_c$ best of these are copied without any change to the next generation, while the rest of them will be replaced by chromosomes produced in crossover and mutation.
   (b) **Apply** the crossover procedure. This procedure creates $p_s \times N_c$ offsprings from the current population. For each pair $(\tilde{z}, \tilde{w})$ of ofssprings two distinct chromosomes (z, w) are chosen from the population. The selection of these chromosomes is performed using tournament selection. The new offsprings $(\tilde{z}, \tilde{w})$ are created using the one - point crossover procedure, which is demonstrated in Figure 2.
   (c) **Perform** the mutation procedure. For each element of every chromosome select a random number $r \in [0, 1]$. If $r \leq p_m$, then the corresponding element is altered randomly.
4. **Local search step.**
   (a) **If** k mod N_I=0 **then**
      i. **Set** $S = \left\{g_{r_1}, g_{r_2}, \ldots, g_{r_{N_T}}\right\}$ a set of randomly selected chromosomes.
      ii. **For** $j = 1, \ldots, N_T$ apply the procedure described in subsection 2.3 on chromosome g_{r_j}.
   (b) **Endif**

5. **Termination check step**.
   (a) **Set** $k = k + 1$
   (b) **If** $k \leq N_g$ goto Fitness Calculation Step, **else**
      i. **Set** g^* the chromosome with the lowest fitness value.
      ii. **Create** the neural network $N^*\left(\vec{x}, \vec{w^*}\right)$ for the this chromosome. The vector $\vec{w^*}$ represents the set of parameters for the chromosome g^*.
      iii. **Apply** the network $N^*\left(\vec{x}, \vec{w^*}\right)$ to the test set and report the error.

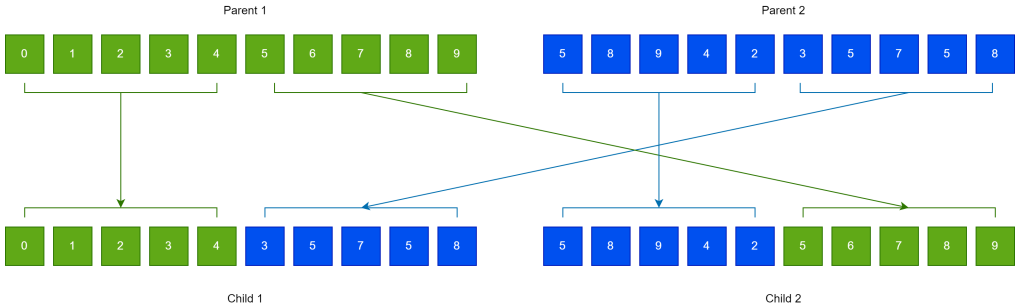

Figure 2. An example of the method of one - point crossover, used in the Grammatical Evolution procedure.

The flowchart for the proposed method is depicted in Figure 3.

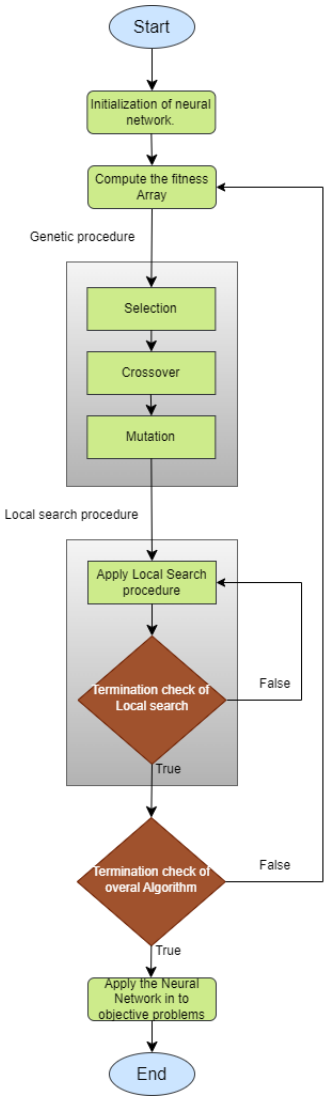

Figure 3. The flowchart of the proposed algorithm.

2.3. The local search procedure

The local search procedure initiates from the vector $\overrightarrow{w}$ that is produced for any given chromosome g using the procedure described in subsection 2.1. The procedure minimizes the error of equation 1 with respect to vector $\overrightarrow{w}$. The minimization is done within a value interval created around the initial point $\overrightarrow{w}$ keeping the network structure intact. The main steps of this procedure are given below:

1. **Set** $d = (n+2)H$ the total number of parameters of the network $N(\overrightarrow{x}, \overrightarrow{w})$.
2. **For** $i = 1, \ldots, d$ **do**
   (a) **Set** $L_i = -F \times |w_i|$, the left bound of the minimization for the parameter i.
   (b) **Set** $R_i = F \times |w_i|$, the right bound of the minimization for the parameter i.
3. **EndFor**
4. **Minimize** the error function of equation 1 with respect to vector $\overrightarrow{w}$ inside the bounding box $\left[\overrightarrow{L}, \overrightarrow{R}\right]$ using a local search procedure $\mathcal{L}(w)$. The BFGS variant of Powell [71] was used a local search procedure in the conducted experiments.

As an example consider a dataset with $n = 2$ and the following constructed neural network:

$$N(\overrightarrow{x}, \overrightarrow{w}) = 2\sigma(1.5x_2 - 2.0) + 2.5\sigma(-1.2x_1 + 3.7) \quad (4)$$

where the number of nodes is $H = 2$. In this case the vector $\overrightarrow{w}$ has the elements $\overrightarrow{w} = (2, 0, 1.5, -2, 2.5, -1.2, 0, 3.7)$. If the parameter F has the value $F = 2$,then the bound vectors $\overrightarrow{L}$ and $\overrightarrow{R}$ are defined as:

$$\begin{aligned} \overrightarrow{L} &= (-4, 0, -3, -4, -5, -2.4, 0, 7.4) \\ \overrightarrow{R} &= (4, 0, 3, 4, 5, 2.4, 0, 7.4) \end{aligned}$$

Hence the minimization method could not change the elements w_2 and w_7 and as a consequence the architecture of the network remains intact.

3. Results

The present research work was tested on a series of data classification and fitting problems, which appeared in recent years in the relevant literature. Also, the proposed method was compared against other well - known machine learning methods from the relevant literature and the results are reported. Also, a series of experiments were performed to verify the sensitivity of the proposed technique with respect to the critical parameters presented earlier. The datasets used in the conducted experiments can be downloaded freely from the following websites:

1. The UCI dataset repository, https://archive.ics.uci.edu/ml/index.php(accessed on 10 September 2024)[72]
2. The Keel repository, https://sci2s.ugr.es/keel/datasets.php(accessed on 10 September 2024)[73].
3. The Statlib URL http://lib.stat.cmu.edu/datasets/(accessed on 10 September 2024).

3.1. Classification datasets

The descriptions of the used datasets are as follows:

1. **Appendictis** a medical dataset, originated in [74].
2. **Australian** dataset [75], used in bank transactions.
3. **Balance** dataset [76], that contains data from psychological experiments.
4. **Circular** dataset, which is an artificial dataset.
5. **Cleveland** dataset, which is a medical dataset [77,78].
6. **Dermatology** dataset [79], a dataset that contains measurements about dermatological deceases.
7. **Ecoli** dataset, a dataset contains measurements about proteins[80].
8. **Fert** dataset, contains data about the relation of sperm concentration and demographic data.
9. **Haberman** dataset, a medical dataset related to breast cancer.
10. **Hayes roth** dataset [81], which is related to a human subjects study.
11. **Heart** dataset [82], a medical dataset used for the prediction of heart diseases.
12. **HeartAttack** dataset, a medical dataset about heart diseases.
13. **HouseVotes** dataset [83], about the votes in the U.S. House of Representatives.
14. **Liverdisorder** dataset [84], a medical dataset related to liver disorders.
15. **Ionosphere** dataset, a climate dataset related to ionosphere experiments [85,86].
16. **Mammographic** dataset [87], a medical dataset for breast cancer.
17. **Parkinsons** dataset, a medical dataset for the detection of Parkinson's disease (PD)[88].
18. **Pima** dataset [89], a medical dataset about the detection of diabetes.
19. **Popfailures** dataset [90], a dataset that contains climate measurements.
20. **Regions2** dataset, a medical dataset related to hepatitis C [91].
21. **Saheart** dataset [92], a medical dataset used for the detection of heart diseases.
22. **Segment** dataset [93], an image processing dataset.
23. **Spiral** dataset, which is an artificial dataset.
24. **Student** dataset [94], related to data collected in Portuguese schools.

25. **Transfusion** dataset[95], a medical dataset used in Blood Transfusion Service Center in Hsin-Chu City in Taiwan.
26. **Wdbc** dataset [96], a medical dataset.
27. **Wine** dataset, used for the detection of quality of wines. [97,98].
28. **Eeg** datasets, a medical dataset related to EEG measurements [99]. From this dataset the following cases were used: Z_F_S, Z_O_N_F_S, ZO_NF_S and ZONF_S.
29. **Zoo** dataset [100], used for the classification of animals in some categories.

3.2. Regression datasets

The used regression datasets are described below:

1. **Abalone** dataset [101], used to predict the age of abalones.
2. **Airfoil** dataset, a dataset provided by NASA [102].
3. **BK** dataset [103], used to predict the points scored in a basketball game.
4. **BL** dataset, related to an electricity experiment.
5. **Baseball** dataset, used to predict the average income of baseball players.
6. **Concrete** dataset [104], which is a civil engineering dataset.
7. **Dee** dataset, that contains measurements about the price of electricity.
8. **FY,** this dataset used to measure the longevity of fruit flies.
9. **HO** dataset, downloaded from the STALIB repository.
10. **Housing** dataset, originated in [105].
11. **Laser** dataset, that contains data from laser experiments
12. **LW** dataset, related to risk factors associated with low weight babies.
13. **MORTGAGE** dataset, an economic dataset.
14. **MUNDIAL,** downloaded from the STALIB repository.
15. **PL** dataset, downloaded from the STALIB repository.
16. **QUAKE** dataset, contains data about the strength of a earthquakes.
17. **REALESTATE,** downloaded from the STALIB repository.
18. **SN** dataset, that contains measurements from an experiment related to trellising and pruning.
19. **Treasury** dataset, an economic dataset.
20. **VE** dataset, downloaded from the STALIB repository.
21. **TZ** dataset, downloaded from the STALIB repository.

3.3. Experimental results

The code used in the experiments was written in ANSI C++ using the optimization environment of Optimus, which is freely available from https://github.com/itsoulos/GlobalOptimus/(accessed on 11 September 2024). All the experiments were conducted and averages were recorded. In every execution, a different seed for the random generator was used. The average classification error as measured on the test is recorded for the classification datasets and the average regression error for the regression datasets. The validation of the conducted experiments was performed using the ten - fold cross validation technique. The experiments were carried out on an AMD Ryzen 5950X with 128GB of RAM, running the Debian Linux operating system. The values for the parameters of the algorithms are shown in Table 1. In all experimental tables, the bold notation is used to mark the method with the lowest classification or regression error.

Table 1. The values used in the experimental parameters.

PARAMETER	MEANING	VALUE
N_c	Number of chromosomes	500
N_g	Maximum number of generations	200
p_s	Crossover rate	0.10
p_m	Mutation rate	0.05
N_T	Number of chromosomes that selected randomly	20
N_I	Number of iterations before local search	10
F	Magnitude of changes from local search	4.0

The experimental results for the classification datasets are shown in Table 2 and the experimental results for the regression datasets our outlined in Table 3. The following applies to the tables of experimental results:

1. The column BFGS denotes the application of the BFGS optimization method to train a neural network with $H = 10$ processing nodes.
2. The column GENETIC stands for the incorporation of a Genetic Algorithm to train a neural network with $H = 10$ processing nodes. The values of the parameters of the genetic algorithm are shown in Table 1.
3. The column NNC represents the application of the original neural network construction model to the provided dataset.
4. The column INNC stands for the application of the current work to the provided dataset.
5. The row AVERAGE denotes the average classification or regression row for all datasets.

Table 2. Experimental results for the classification datasets of this article. Numbers in cells represent average classification error as measured on the corresponding test set.

DATASET	BFGS	GENETIC	NNC	INNC
APPENDICITIS	18.00%	24.40%	**13.70%**	14.70%
AUSTRALIAN	38.13%	36.64%	**14.51%**	14.80%
BALANCE	8.64%	**8.36%**	22.11%	8.66%
CIRCULAR	6.08%	**5.13%**	13.64%	5.32%
CLEVELAND	77.55%	57.21%	50.10%	**47.93%**
DERMATOLOGY	52.92%	**16.60%**	25.06%	20.89%
ECOLI	69.52%	54.67%	**47.82%**	48.21%
FERT	23.20%	28.50%	**19.00%**	20.50%
HABERMAN	29.34%	28.66%	28.03%	**26.70%**
HAYES-ROTH	37.33%	56.18%	35.93%	**31.77%**
HEART	39.44%	26.41%	15.78%	**14.74%**
HEARTATTACK	46.67%	29.03%	**19.33%**	20.43%
HOUSEVOTES	7.13%	7.00%	3.65%	**3.26%**
IONOSPHERE	15.29%	15.14%	**11.12%**	11.92%
LIVERDISORDER	42.59%	37.09%	33.71%	**31.77%**
MAMMOGRAPHIC	17.24%	19.88%	17.78%	**15.81%**
PARKINSONS	27.58%	16.58%	**12.21%**	12.53%
PIMA	35.59%	34.21%	27.99%	**24.00%**
POPFAILURES	5.24%	**4.17%**	6.74%	6.44%
REGIONS2	36.28%	33.53%	25.52%	**23.18%**
SAHEART	37.48%	34.85%	30.52%	**28.09%**
SEGMENT	68.97%	46.30%	54.99%	**43.12%**
SPIRAL	47.99%	47.67%	48.39%	**43.99%**
STUDENT	7.14%	5.61%	5.78%	**4.55%**
TRANSFUSION	25.80%	25.84%	25.34%	**23.43%**
WDBC	29.91%	7.87%	6.95%	**4.41%**
WINE	59.71%	22.88%	14.35%	**9.77%**
Z_F_S	39.37%	24.60%	14.17%	**8.53%**
Z_O_N_F_S	65.67%	64.81%	49.18%	**38.58%**
ZO_NF_S	43.04%	21.54%	14.14%	**6.84%**
ZONF_S	15.62%	4.36%	3.14%	**2.52%**
ZOO	10.70%	9.50%	9.20%	**7.20%**
AVERAGE	**33.91%**	**26.73%**	**22.50%**	**19.52%**

Table 3. Experimental results for the regression datasets presented in this manuscript. Numbers in cells denote average regression error as calculated on the corresponding test set.

DATASET	BFGS	GENETIC	NNC	INNC
ABALONE	5.69	7.17	5.05	**4.33**
AIRFOIL	0.003	0.003	0.003	**0.002**
BASEBALL	119.63	64.60	59.85	**48.42**
BK	0.36	0.26	2.32	**0.07**
BL	1.09	2.23	0.021	**0.002**
CONCRETE	0.066	0.01	0.008	**0.005**
DEE	2.36	1.01	0.26	**0.23**
HO	0.62	0.37	0.017	**0.01**
HOUSING	97.38	43.26	26.35	**16.01**
FY	0.19	0.65	0.058	**0.042**
LASER	0.015	0.59	0.024	**0.005**
LW	2.98	0.54	**0.011**	0.012
MORTGAGE	8.23	0.40	0.30	**0.026**
MUNDIAL	6.48	1.22	4.47	**0.034**
PL	0.29	0.28	0.045	**0.022**
QUAKE	0.42	0.12	0.045	**0.04**
REALESTATE	128.94	81.19	76.78	**70.99**
SN	3.89	0.40	0.026	**0.023**
TREASURY	9.91	2.93	0.47	**0.066**
TZ	3.27	5.38	5.04	**0.03**
VE	1.92	2.43	6.61	**0.025**
AVERAGE	**18.75**	**10.24**	**8.94**	**6.69**

The original technique of constructing artificial neural networks has significantly lower classification or regression errors than the other two techniques and the proposed method significantly improves the performance of this technique in the majority of the datasets. The percentage improvement in error is even greater in regression problems. In some cases the improvement exceeds 50% in the test error. This effect is evident in the box plots for classification and regression errors outlined in Figures 4 and 5 respectively.

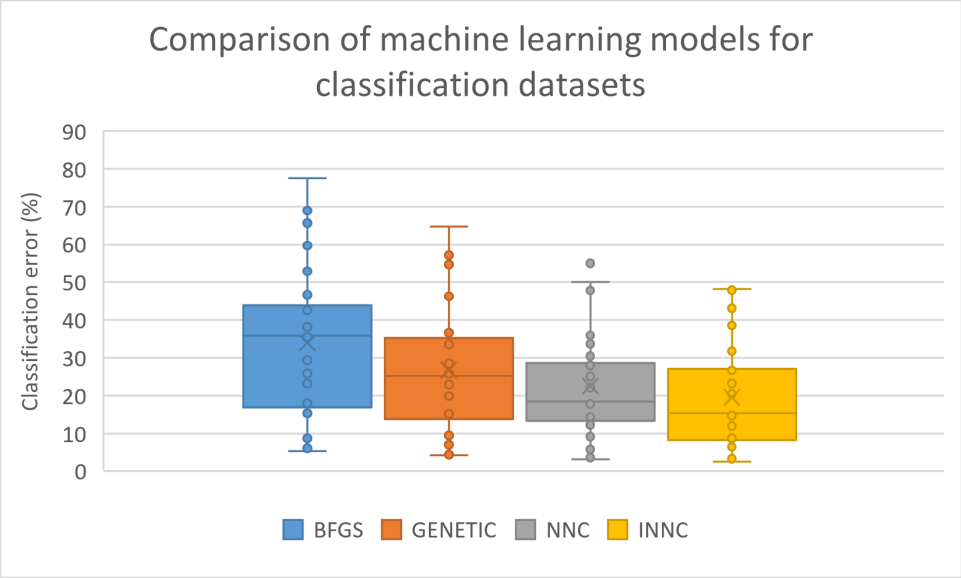

Figure 4. Box plot for the comparison between the machine learning methods applied on the classification datasets.

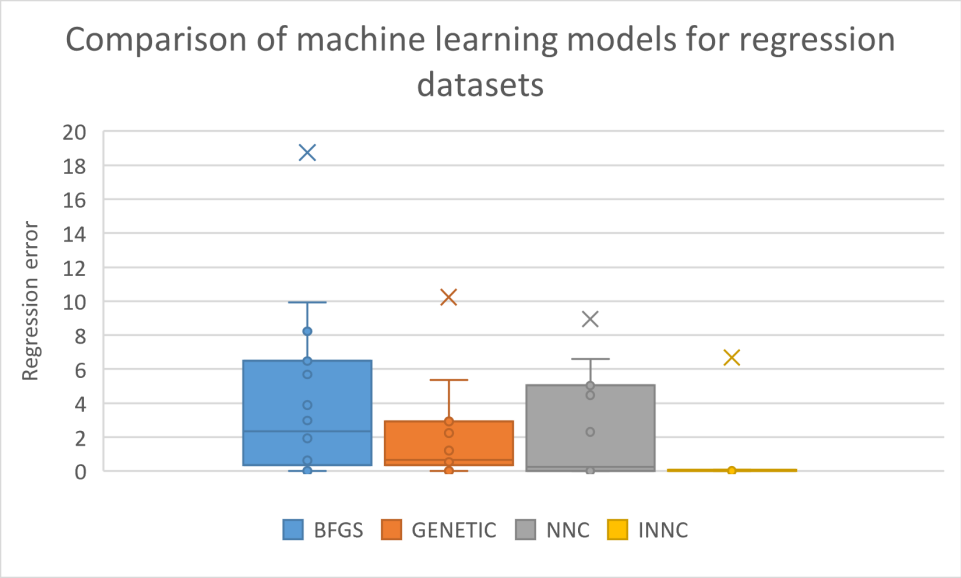

Figure 5. Box plot for the comparison between the machine learning methods applied on the regression datasets.

Furthermore, the same reduction in test error is validated from the statistical tests performed for classification and regression problems. These tests are depicted in Figures 6 and 7.

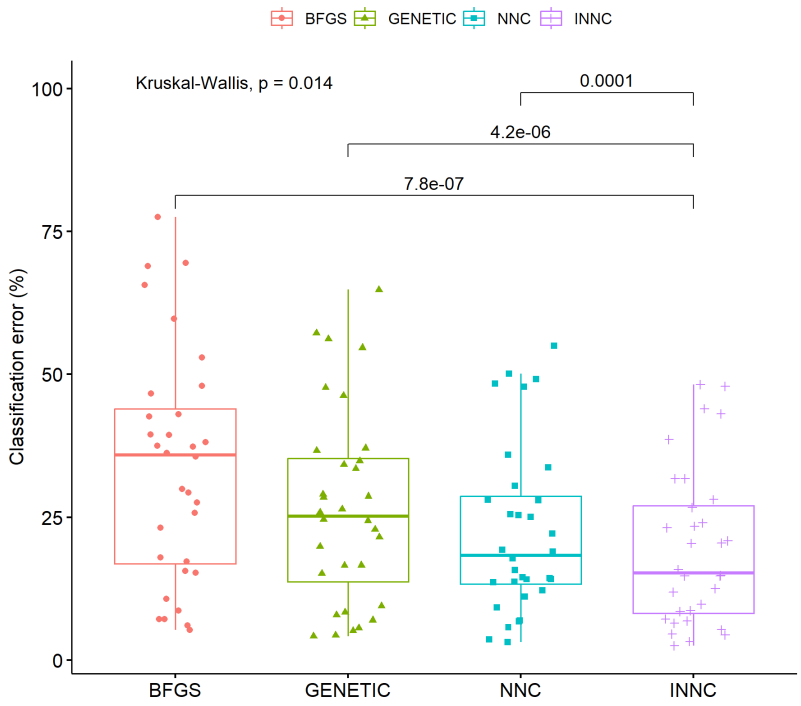

Figure 6. Statistical comparison between the machine learning models for the classification datasets.

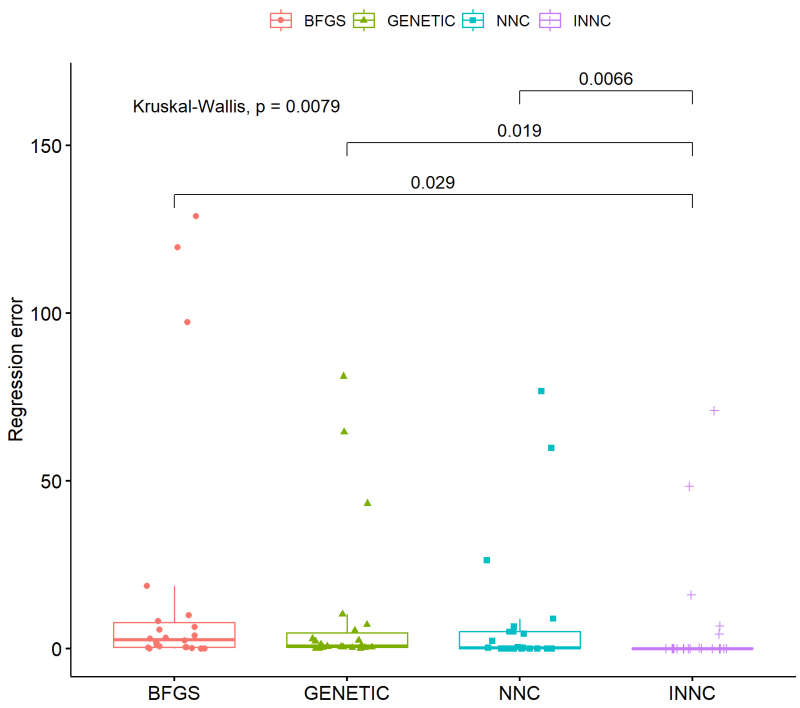

Figure 7. Statistical comparison between the machine learning models for the regression datasets.

3.3.1. Experiments with the parameter N_I

To verify the robustness of the proposed technique as well as its sensitivity to parameter changes, an additional experiment was performed in which the critical N_I parameter was varied from 5 to 20. This parameter determines the number of generations intervening before the local optimization method is executed on randomly selected chromosomes. The results from this experiment for the classification datasets are depicted in Table 4 and the results for the regression datasets in Table 5.

Table 4. Experimental results with different values for the parameter N_I of the proposed method. The method was applied on the classification datasets.

DATASET	**INNC**$(N_I = 5)$	**INNC**$(N_I = 10)$	**INNC**$(N_I = 20)$
APPENDICITIS	14.50%	14.70%	**14.20%**
AUSTRALIAN	**14.48%**	14.80%	14.58%
BALANCE	**7.63%**	8.66%	11.71%
CIRCULAR	**5.25%**	5.32%	5.87%
CLEVELAND	48.41%	**47.93%**	49.66%
DERMATOLOGY	**19.66%**	20.89%	23.11%
ECOLI	**47.39%**	48.21%	48.09%
FERT	20.80%	**20.50%**	20.90%
HABERMAN	26.87%	**26.70%**	27.27%
HAYES-ROTH	**31.69%**	31.77%	35.92%
HEART	14.85%	**14.74%**	15.56%
HEARTATTACK	20.10%	20.43%	**20.07%**
HOUSEVOTES	3.87%	**3.26%**	3.78%
IONOSPHERE	11.51%	11.92%	**11.09%**
LIVERDISORDER	**31.35%**	31.77%	31.85%
MAMMOGRAPHIC	16.25%	**15.81%**	16.06%
PARKINSONS	13.16%	**12.53%**	12.58%
PIMA	**23.95%**	24.00%	24.67%
POPFAILURES	**6.06%**	6.44%	6.48%
REGIONS2	23.36%	**23.18%**	24.21%
SAHEART	**27.68%**	28.09%	29.41%
SEGMENT	43.79%	**43.12%**	46.91%
SPIRAL	**43.63%**	43.99%	45.15%
STUDENT	**3.65%**	4.55%	3.88%
TRANSFUSION	**22.62%**	23.43%	23.89%
WDBC	**4.41%**	4.41%	5.46%
WINE	10.18%	**9.77%**	12.00%
Z_F_S	**7.70%**	8.53%	9.63%
Z_O_N_F_S	**36.64%**	38.58%	41.08%
ZO_NF_S	**6.84%**	6.84%	6.90%
ZONF_S	**2.24%**	2.52%	2.66%
ZOO	**6.30%**	7.20%	7.70%
AVERAGE	**19.28%**	**19.52%**	**20.39%**

Table 5. Experimental results with different values for the parameter N_I of the proposed method. The method was applied on the regression datasets.

DATASET	**INNC**($N_I = 5$)	**INNC**($N_I = 10$)	**INNC**($N_I = 20$)
ABALONE	4.38	**4.33**	4.45
AIRFOIL	0.002	**0.002**	0.002
BASEBALL	**47.50**	48.42	49.99
BK	0.64	**0.07**	1.86
BL	0.014	**0.002**	0.003
CONCRETE	**0.005**	0.005	0.005
DEE	**0.22**	0.23	0.23
HO	**0.01**	0.01	0.011
HOUSING	21.09	**16.01**	16.96
FY	0.046	**0.042**	0.19
LASER	**0.004**	0.005	0.005
LW	**0.011**	0.012	0.011
MORTGAGE	**0.02**	0.026	0.03
MUNDIAL	0.031	0.034	**0.029**
PL	**0.022**	0.022	0.022
QUAKE	0.045	0.04	**0.037**
REALESTATE	**68.85**	70.99	70.59
SN	**0.023**	0.023	0.024
TREASURY	**0.063**	0.066	0.073
TZ	**0.029**	0.03	0.031
VE	0.028	**0.025**	0.116
AVERAGE	**6.81**	**6.69**	**6.89**

The method appears to have similar results for each variation of the critical N_I parameter. In some cases the error is smaller for low values of this parameter but not to a great extent. This effect is also evident in the box plot presented for the classification datasets in Figure 8 as well as in the statistical comparison of Figure 9.

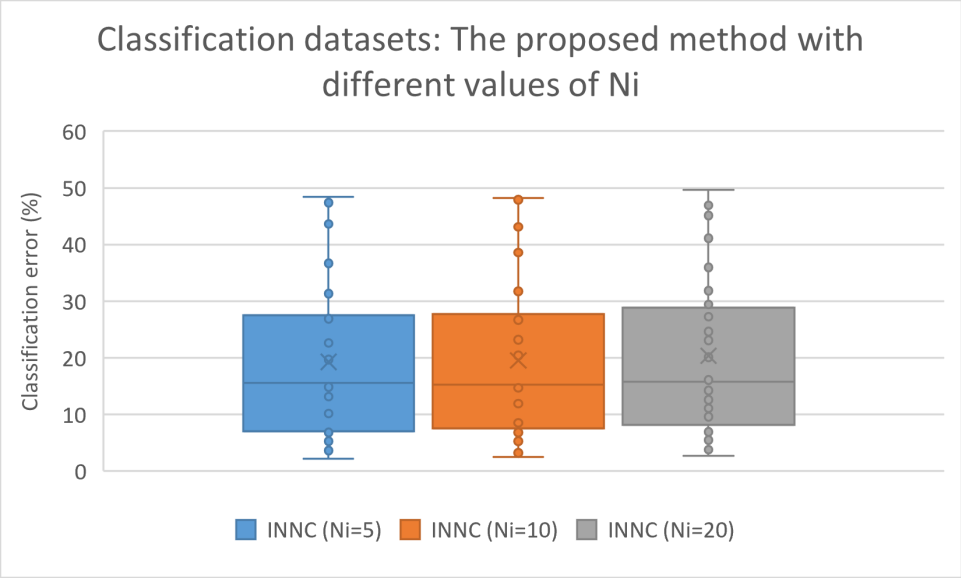

Figure 8. Box plot for the experiments with the parameter N_I and the proposed method. The method was applied on the classification datasets.

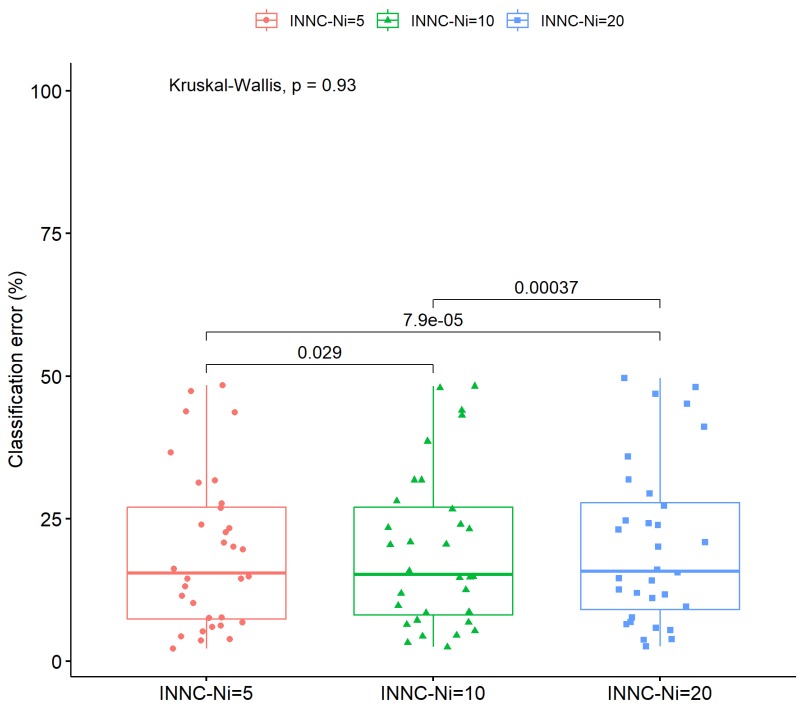

Figure 9. Statistical comparison for the experiment with the proposed method and different values of parameter N_I. The method was applied on the classification datasets.

3.3.2. Experiments with the parameter F

Another important parameter for the proposed method is the parameter F. This parameter controls the range of changes that the local optimization method can cause on randomly selected chromosomes. In this experiment, the parameter F changed from 1.5 to 8.0 and the results for the classification datasets are listed in Table 6 while the experimental results for the regression datasets are presented in Table 7.

Table 6. Experimental results with different values for the parameter F of the proposed method. The method was applied on the classification datasets.

DATASET	**INNC**$(F = 1.5)$	**INNC**$(F = 2.0)$	**INNC**$(F = 4.0)$	**INNC**$(F = 8.0)$
APPENDICITIS	14.80%	**14.20%**	14.70%	15.00%
AUSTRALIAN	14.58%	14.56%	14.80%	**14.54%**
BALANCE	**8.52%**	8.68%	8.66%	9.35%
CIRCULAR	5.92%	5.46%	**5.32%**	5.43%
CLEVELAND	48.35%	**47.38%**	47.93%	49.62%
DERMATOLOGY	**19.80%**	20.09%	20.89%	21.66%
ECOLI	**47.76%**	48.30%	48.21%	48.42%
FERT	21.20%	21.20%	**20.50%**	20.90%
HABERMAN	26.43%	**25.93%**	26.70%	26.70%
HAYES-ROTH	32.15%	**31.31%**	31.77%	35.69%
HEART	14.81%	15.41%	**14.74%**	15.41%
HEARTATTACK	21.20%	**19.90%**	20.43%	20.40%
HOUSEVOTES	3.35%	3.65%	**3.26%**	3.65%
IONOSPHERE	11.31%	**10.40%**	11.92%	11.23%
LIVERDISORDER	32.09%	**31.21%**	31.77%	33.18%
MAMMOGRAPHIC	16.15%	16.05%	**15.81%**	16.07%
PARKINSONS	12.63%	**12.47%**	12.53%	13.05%
PIMA	**23.58%**	23.80%	24.00%	23.79%
POPFAILURES	**5.85%**	6.29%	6.44%	6.04%
REGIONS2	24.08%	23.48%	**23.18%**	25.25%
SAHEART	28.41%	28.83%	**28.09%**	28.30%
SEGMENT	45.12%	43.37%	**43.12%**	43.93%
SPIRAL	**43.40%**	43.72%	43.99%	44.61%
STUDENT	4.25%	**4.15%**	4.55%	4.52%
TRANSFUSION	23.09%	**22.93%**	23.43%	23.05%
WDBC	5.27%	5.30%	**4.41%**	4.84%
WINE	10.53%	10.24%	**9.77%**	11.59%
Z_F_S	9.20%	**7.57%**	8.53%	8.63%
Z_O_N_F_S	40.60%	38.70%	**38.58%**	39.78%
ZO_NF_S	6.90%	7.68%	6.84%	**6.56%**
ZONF_S	2.64%	2.72%	**2.52%**	2.88%
ZOO	8.60%	7.40%	**7.20%**	8.90%
AVERAGE	**19.77%**	**19.45%**	**19.52%**	**20.09%**

Table 7. Experimental results with different values for the parameter F of the proposed method. The method was applied on the regression datasets.

DATASET	INNC($F = 1.5$)	INNC($F = 2.0$)	INNC($F = 4.0$)	INNC($F = 8.0$)
ABALONE	4.49	4.43	**4.33**	4.66
AIRFOIL	0.002	0.002	**0.002**	0.002
BASEBALL	**48.31**	50.38	48.42	48.52
BK	0.06	0.21	0.07	**0.02**
BL	0.003	0.004	**0.002**	0.003
CONCRETE	**0.005**	0.005	0.005	0.005
DEE	**0.23**	0.23	0.23	0.23
HO	**0.01**	0.01	0.01	0.01
HOUSING	16.91	16.34	**16.01**	16.96
FY	0.042	0.043	**0.042**	0.043
LASER	0.005	0.005	0.005	**0.004**
LW	**0.011**	0.012	0.012	0.012
MORTGAGE	0.027	0.025	0.026	**0.024**
MUNDIAL	0.035	**0.033**	0.034	0.23
PL	**0.022**	0.022	0.022	0.022
QUAKE	**0.036**	0.037	0.04	0.036
REALESTATE	72.40	71.16	70.99	**70.76**
SN	0.024	0.024	**0.023**	0.024
TREASURY	0.07	0.068	**0.066**	0.068
TZ	**0.029**	0.031	0.03	0.031
VE	0.032	0.03	**0.025**	0.028
AVERAGE	**6.80**	**6.81**	**6.69**	**6.75**

Once again, there are no significant differences in the performance of the proposed technique as the critical factor F varies. In the case of the classification data, however, there is a small increase in the classification error as this factor increases, which may be due to the fact that, as we move away from the solution created by the method of creating neural networks, the performance of the method decreases. Also, the box plot for the classification datasets is depicted in Figure 10.

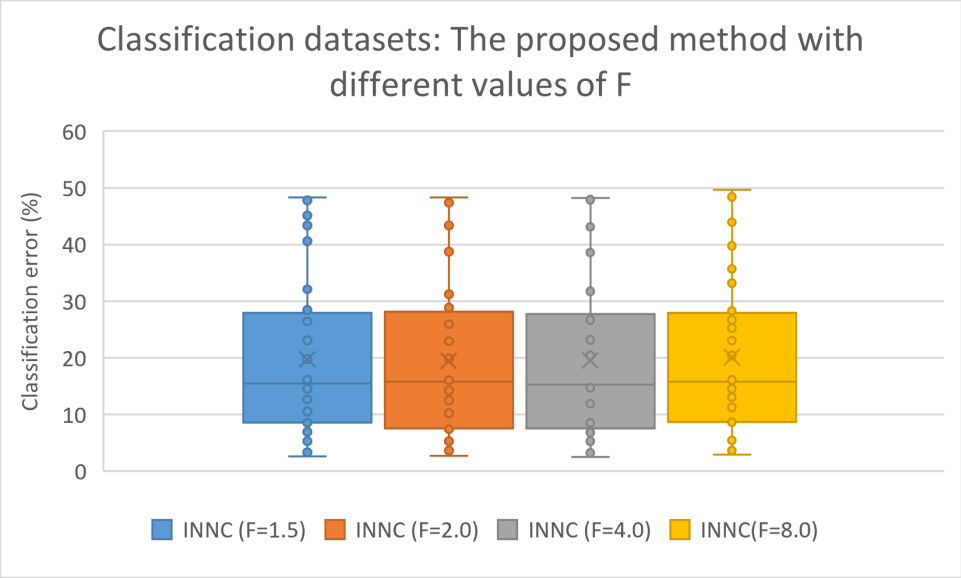

Figure 10. Box plot for the experiments with the proposed method and different values of the parameter F. The method was applied on the classification datasets.

3.3.3. Experiments with the used local search optimizer

An important issue of the proposed method is the selection of the local search optimizer, that will be applied periodically to randomly selected chromosomes. In the current work, a BFGS variant [71] was chosen since it can satisfactorily handle the constraints placed on the network parameters for the optimization. However, an additional experiment was executed using different local optimization techniques and the results for the classification datasets are presented in Table 8 and the results for the regression datasets are shown in Table 9. The columns in these result tables have the following meanings:

1. The column ADAM denotes the incorporation of the ADAM local optimization method [106] in the current technique.
2. The column LBFGS stands for the usage of Limited Memory BFGS (L-BFGS) method [107] in the current work.
3. The column BFGS represents the incorporation of the BFGS variant of Powell [71]as the local search procedure.

Table 8. Experimental results with different local optimization techniques in the proposed method. The method was applied on the classification datasets.

DATASET	ADAM	LBFGS	BFGS
APPENDICITIS	**14.60%**	14.60%	14.70%
AUSTRALIAN	**14.78%**	15.29%	14.80%
BALANCE	16.89%	11.47%	**8.66%**
CIRCULAR	10.79%	8.15%	**5.32%**
CLEVELAND	49.10%	48.45%	**47.93%**
DERMATOLOGY	22.32%	23.03%	**20.89%**
ECOLI	**48.15%**	49.48%	48.21%
FERT	**18.10%**	20.10%	20.50%
HABERMAN	**26.27%**	26.63%	26.70%
HAYES-ROTH	36.23%	36.31%	**31.77%**
HEART	15.71%	15.00%	**14.74%**
HEARTATTACK	21.33%	20.07%	20.43%
HOUSEVOTES	3.39%	3.39%	**3.26%**
IONOSPHERE	**10.88%**	11.29%	11.92%
LIVERDISORDER	31.80%	33.18%	**31.77%**
MAMMOGRAPHIC	16.71%	16.30%	**15.81%**
PARKINSONS	**12.10%**	12.32%	12.53%
PIMA	25.49%	24.97%	**24.00%**
POPFAILURES	**6.20%**	6.31%	6.44%
REGIONS2	24.20%	24.06%	**23.18%**
SAHEART	28.87%	28.96%	**28.09%**
SEGMENT	50.16%	46.63%	**43.12%**
SPIRAL	46.33%	45.24%	43.99%
STUDENT	4.05%	**3.98%**	4.55%
TRANSFUSION	24.74%	24.56%	**23.43%**
WDBC	6.30%	5.66%	**4.41%**
WINE	11.47%	10.06%	**9.77%**
Z_F_S	12.33%	10.92%	**8.53%**
Z_O_N_F_S	41.40%	45.77%	**38.58%**
ZO_NF_S	11.26%	9.83%	**6.84%**
ZONF_S	2.64%	**2.35%**	2.52%
ZOO	8.50%	8.70%	**7.20%**
AVERAGE	**21.03%**	**20.72%**	**19.52%**

Table 9. Experimental results with local optimization techniques in the proposed method. The method was applied on the regression datasets.

DATASET	ADAM	LBFGS	BFGS
ABALONE	4.73	4.49	**4.33**
AIRFOIL	0.003	0.003	**0.002**
BASEBALL	53.98	51.88	**48.42**
BK	**0.05**	0.08	0.07
BL	0.013	0.007	**0.002**
CONCRETE	0.006	0.006	**0.005**
DEE	0.24	**0.23**	0.23
HO	0.012	0.012	**0.01**
HOUSING	20.20	20.49	**16.01**
FY	**0.042**	0.042	0.042
LASER	0.012	**0.005**	0.005
LW	**0.011**	0.011	0.012
MORTGAGE	0.068	0.084	**0.026**
MUNDIAL	0.033	**0.029**	0.034
PL	0.031	0.023	**0.022**
QUAKE	**0.036**	0.037	0.04
REALESTATE	74.66	74.49	**70.99**
SN	0.025	0.024	**0.023**
TREASURY	0.114	0.084	**0.066**
TZ	**0.03**	0.08	0.03
VE	**0.024**	0.132	0.025
AVERAGE	**7.35**	**7.25**	**6.69**

The BFGS method achieved lower values for the test error in the majority of cases and this is also evident in the Figure 11, where a box plot is depicted for the classification datasets.

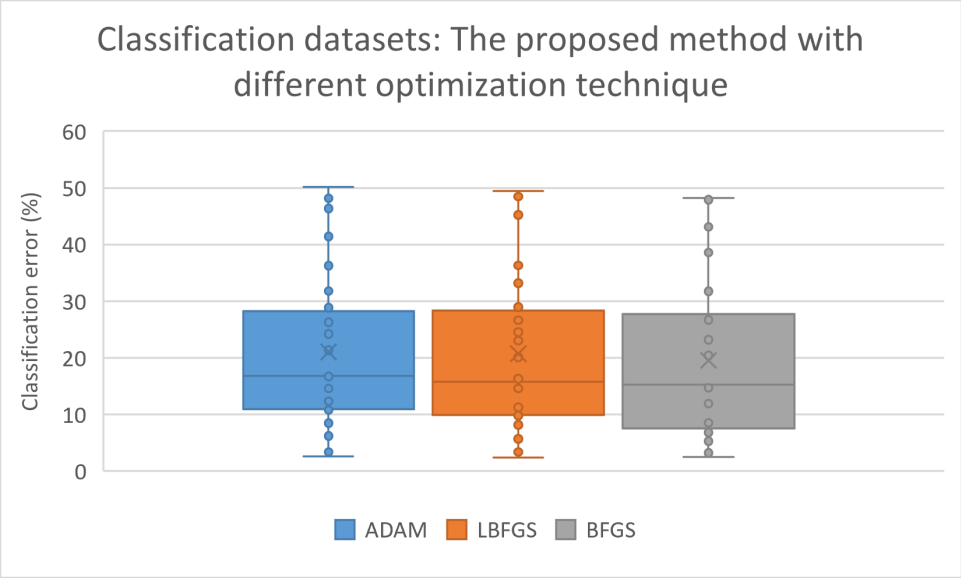

Figure 11. Box plot for the experiment with the proposed method and different local optimization methods. The proposed method was applied on the classification datasets.

4. Conclusions

An extension of the artificial neural network construction technique was presented in the present work, in which the periodic application of a local optimization method to randomly selected chromosomes was introduced. The application of the local optimization method is done in such a way as not to alter the architecture of the neural network

constructed by Grammatical Evolution. The proposed modified method was applied on a series of benchmark datasets found in the relevant literature and, judging from the experimental results, it reduced significantly the test error of the original method in the majority of datasets.

Moreover, to establish the stability of the proposed technique, a series of experiments were performed in which a number of critical parameters were varied over a range of values. After the completion of these experiments, it became clear that there is no significant difference in the effectiveness of the proposed method, even if these critical parameters change significantly from execution to execution. The only case where a significant difference in the effectiveness of the proposed technique was found was when different local optimization techniques were used, where the BFGS variant appeared to achieve the best results in the majority of cases.

Nevertheless, one major drawback of the proposed method is the additional execution time required from the execution of the local search optimization techniques. Since the Grammatical Evolution procedure is a modified genetic algorithm, the generated artificial neural networks are independent of themselves, parallel programming techniques may be used in order to speed up the process, such as the usage of MPI [108] or the OpenMP library [109].

Author Contributions: V.C. and I.G.T. conducted the experiments, employing several datasets and provided the comparative experiments. D.T., A.T. and V.C. performed the statistical analysis and prepared the manuscript. 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.

Institutional Review Board Statement: Not applicable.

Institutional Review Board Statement: Not applicable.

Acknowledgments: 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).

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

References

1. C. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
2. G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of Control Signals and Systems **2**, pp. 303-314, 1989.
3. P. Baldi, K. Cranmer, T. Faucett et al, Parameterized neural networks for high-energy physics, Eur. Phys. J. C **76**, 2016.
4. 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
5. G. Carleo,M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science **355**, pp. 602-606, 2017.
6. Y. Shirvany, M. Hayati, R. Moradian, Multilayer perceptron neural networks with novel unsupervised training method for numerical solution of the partial differential equations, Applied Soft Computing **9**, pp. 20-29, 2009.
7. A. Malek, R. Shekari Beidokhti, Numerical solution for high order differential equations using a hybrid neural network—Optimization method, Applied Mathematics and Computation **183**, pp. 260-271, 2006.
8. A. Kumar Yadav, S.S. Chandel, Solar radiation prediction using Artificial Neural Network techniques: A review, Renewable and Sustainable Energy Reviews **33**, pp. 772-781, 2014.
9. A. Topuz, Predicting moisture content of agricultural products using artificial neural networks, Advances in Engineering Software **41**, pp. 464-470, 2010.
10. A. Escamilla-García, G.M. Soto-Zarazúa, M. Toledano-Ayala, E. Rivas-Araiza, A. Gastélum-Barrios, Abraham,Applications of Artificial Neural Networks in Greenhouse Technology and Overview for Smart Agriculture Development, Applied Sciences **10**, Article number 3835, 2020.
11. Lin Shen, Jingheng Wu, and Weitao Yang, Multiscale Quantum Mechanics/Molecular Mechanics Simulations with Neural Networks, Journal of Chemical Theory and Computation **12**, pp. 4934-4946, 2016.

12. Sergei Manzhos, Richard Dawes, Tucker Carrington, Neural network-based approaches for building high dimensional and quantum dynamics-friendly potential energy surfaces, Int. J. Quantum Chem. **115**, pp. 1012-1020, 2015.
13. Jennifer N. Wei, David Duvenaud, and Alán Aspuru-Guzik, Neural Networks for the Prediction of Organic Chemistry Reactions, ACS Central Science **2**, pp. 725-732, 2016.
14. G. Li, J. Shi, On comparing three artificial neural networks for wind speed forecasting, Applied Energy **87**, pp. 2313-2320, 2010.
15. Lukas Falat and Lucia Pancikova, Quantitative Modelling in Economics with Advanced Artificial Neural Networks, Procedia Economics and Finance **34**, pp. 194-201, 2015.
16. Mohammad Namazi, Ahmad Shokrolahi, Mohammad Sadeghzadeh Maharluie, Detecting and ranking cash flow risk factors via artificial neural networks technique, Journal of Business Research **69**, pp. 1801-1806, 2016.
17. G. Tkacz, Neural network forecasting of Canadian GDP growth, International Journal of Forecasting **17**, pp. 57-69, 2001.
18. 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.
19. Ronadl Bartzatt, Prediction of Novel Anti-Ebola Virus Compounds Utilizing Artificial Neural Network (ANN), Chemistry Faculty Publications **49**, pp. 16-34, 2018.
20. D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning representations by back-propagating errors, Nature **323**, pp. 533 - 536 , 1986.
21. K. Vora, S. Yagnik, A survey on backpropagation algorithms for feedforward neural networks, International Journal of Engineering Development and Research **1**, pp. 193-197, 2014.
22. 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.
23. C. Igel, M. Hüsken, Empirical evaluation of the improved Rprop learning algorithms, Neurocomputing **50**, pp. 105-123, 2003.
24. 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.
25. A. Yamazaki, M. C. P. de Souto,T. B. Ludermir, Optimization of neural network weights and architectures for odor recognition using simulated annealing, In: Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 **1**, pp. 547-552 , 2002.
26. C.L. Kuo, E.E. Kuruoglu, W.K.V. Chan, Neural Network Structure Optimization by Simulated Annealing, Entropy **24**, 348, 2022.
27. 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
28. Z. Liu, A. Liu, C. Wang, Z. Niu, Evolving neural network using real coded genetic algorithm (GA) for multispectral image classification, Future Generation Computer Systems **20**, pp. 1119-1129, 2004.
29. C. Zhang, H. Shao and Y. Li, Particle swarm optimisation for evolving artificial neural network, IEEE International Conference on Systems, Man, and Cybernetics, , pp. 2487-2490, 2000.
30. J. lonen, J.K. Kamarainen, J. Lampinen, Differential Evolution Training Algorithm for Feed-Forward Neural Networks, Neural Processing Letters **17**, pp. 93–105, 2003.
31. K.M. Salama, A.M. Abdelbar, Learning neural network structures with ant colony algorithms, Swarm Intell **9**, pp. 229–265, 2015.
32. S. Mirjalili, How effective is the Grey Wolf optimizer in training multi-layer perceptrons, Appl Intell **43**, pp. 150–161, 2015.
33. I. Aljarah, H. Faris, S. Mirjalili, Optimizing connection weights in neural networks using the whale optimization algorithm, Soft Comput **22**, pp. 1–15, 2018.
34. K-S Oh, K. Jung, GPU implementation of neural networks, Pattern Recognition **37**, pp. 1311-1314, 2004.
35. A.A. Huqqani, E.Schikuta, S. Ye, P. Chen, Multicore and GPU Parallelization of Neural Networks for Face Recognition, Procedia Computer Science **18**, pp. 349-358, 2013.
36. M. Zhang, K. Hibi, J. Inoue, GPU-accelerated artificial neural network potential for molecular dynamics simulation, Computer Physics Communications **285**, 108655, 2023.
37. T.M. Varnava, A.J.Meade, An initialization method for feedforward artificial neural networks using polynomial bases, Advances in Adaptive Data Analysis **3**, pp. 385-400, 2011.
38. I. Ivanova, M. Kubat, Initialization of neural networks by means of decision trees, Knowledge-Based Systems **8**, pp. 333-344, 1995.
39. S.S. Sodhi, P. Chandra, Interval based Weight Initialization Method for Sigmoidal Feedforward Artificial Neural Networks, AASRI Procedia **6**, pp. 19-25, 2014.
40. K. Chumachenko, A. Iosifidis, M. Gabbouj, Feedforward neural networks initialization based on discriminant learning, Neural Networks **146**, pp. 220-229, 2022.
41. Q. Chen, W. Hao, J. He, A weight initialization based on the linear product structure for neural networks, Applied Mathematics and Computation **415**, 126722, 2022.
42. J. Arifovic, R. Gençay, Using genetic algorithms to select architecture of a feedforward artificial neural network, Physica A: Statistical Mechanics and its Applications **289**, pp. 574-594, 2001.
43. P.G. Benardos, G.C. Vosniakos, Optimizing feedforward artificial neural network architecture, Engineering Applications of Artificial Intelligence **20**, pp. 365-382, 2007.
44. B.A. Garro, R.A. Vázquez, Designing Artificial Neural Networks Using Particle Swarm Optimization Algorithms, Computational Intelligence and Neuroscience, 369298, 2015.

45. B. Baker, O. Gupta, N. Naik, R. Raskar, Designing neural network architectures using reinforcement learning. arXiv preprint arXiv:1611.02167, 2016.
46. M. M. Islam, M. A. Sattar, M. F. Amin, X. Yao, K. Murase, A New Adaptive Merging and Growing Algorithm for Designing Artificial Neural Networks, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **39**, pp. 705-722, 2009.
47. M. O'Neill, C. Ryan, Grammatical evolution, IEEE Trans. Evol. Comput. **5,**pp. 349–358, 2001.
48. I.G. Tsoulos, D. Gavrilis, E. Glavas, Neural network construction and training using grammatical evolution, Neurocomputing **72**, pp. 269-277, 2008.
49. G.V. Papamokos, I.G. Tsoulos, I.N. Demetropoulos, E. Glavas, Location of amide I mode of vibration in computed data utilizing constructed neural networks, Expert Systems with Applications **36**, pp. 12210-12213, 2009.
50. I.G. Tsoulos, D. Gavrilis, E. Glavas, Solving differential equations with constructed neural networks, Neurocomputing **72**, pp. 2385-2391, 2009.
51. I.G. Tsoulos, G. Mitsi, A. Stavrakoudis, S. Papapetropoulos, Application of Machine Learning in a Parkinson's Disease Digital Biomarker Dataset Using Neural Network Construction (NNC) Methodology Discriminates Patient Motor Status, Frontiers in ICT 6, 10, 2019.
52. 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.
53. E.I. Toki, J. Pange, G. Tatsis, K. Plachouras, I.G. Tsoulos, Utilizing Constructed Neural Networks for Autism Screening, Applied Sciences **14**, 3053, 2024.
54. I.G. Tsoulos, A.Tzallas, D. Tsalikakis, NNC: A tool based on Grammatical Evolution for data classification and differential equation solving, SoftwareX **10**, 100297, 2019.
55. 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.
56. 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.
57. 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.
58. C. Ryan, M. O'Neill, J.J. Collins, Grammatical evolution: Solving trigonometric identities, proceedings of Mendel. Vol. 98. 1998.
59. 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.
60. I. Dempsey, M.O' Neill, A. Brabazon, Constant creation in grammatical evolution, International Journal of Innovative Computing and Applications **1** , pp 23–38, 2007.
61. 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.
62. 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.
63. 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.
64. 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.
65. 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.
66. 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.
67. F. Castejón, E.J. Carmona, Automatic design of analog electronic circuits using grammatical evolution, Applied Soft Computing **62**, pp. 1003-1018, 2018.
68. L. Araujo, J. Martinez-Romo, A. Duque, Discovering taxonomies in Wikipedia by means of grammatical evolution. Soft Comput **22**, pp. 2907–2919, 2018.
69. C. Martín, D. Quintana, P. Isasi, Grammatical Evolution-based ensembles for algorithmic trading, Applied Soft Computing **84**, 105713, 2019.
70. Moore, J.H., Sipper, M. (2018). Grammatical Evolution Strategies for Bioinformatics and Systems Genomics. In: Ryan, C., O'Neill, M., Collins, J. (eds) Handbook of Grammatical Evolution. Springer, Cham. https://doi.org/10.1007/978-3-319-78717-6_16
71. M.J.D Powell, A Tolerant Algorithm for Linearly Constrained Optimization Calculations, Mathematical Programming **45**, pp. 547-566, 1989.
72. M. Kelly, R. Longjohn, K. Nottingham, The UCI Machine Learning Repository. 2023. Available online: https://archive.ics.uci.edu (accessed on 18 February 2024).

73. 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.
74. 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.
75. J.R. Quinlan, Simplifying Decision Trees. International Journal of Man-Machine Studies **27**, pp. 221-234, 1987.
76. T. Shultz, D. Mareschal, W. Schmidt, Modeling Cognitive Development on Balance Scale Phenomena, Machine Learning **16**, pp. 59-88, 1994.
77. Z.H. Zhou,Y. Jiang, NeC4.5: neural ensemble based C4.5," in IEEE Transactions on Knowledge and Data Engineering **16**, pp. 770-773, 2004.
78. R. Setiono , W.K. Leow, FERNN: An Algorithm for Fast Extraction of Rules from Neural Networks, Applied Intelligence **12**, pp. 15-25, 2000.
79. 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.
80. P. Horton, K.Nakai, A Probabilistic Classification System for Predicting the Cellular Localization Sites of Proteins, In: Proceedings of International Conference on Intelligent Systems for Molecular Biology **4**, pp. 109-15, 1996.
81. 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.
82. I. Kononenko, E. Šimec, M. Robnik-Šikonja, Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF, Applied Intelligence **7**, pp. 39–55, 1997
83. 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.
84. J. Garcke, M. Griebel, Classification with sparse grids using simplicial basis functions, Intell. Data Anal. **6**, pp. 483-502, 2002.
85. J.G. Dy , C.E. Brodley, Feature Selection for Unsupervised Learning, The Journal of Machine Learning Research **5**, pp 845–889, 2004.
86. S. J. Perantonis, V. Virvilis, Input Feature Extraction for Multilayered Perceptrons Using Supervised Principal Component Analysis, Neural Processing Letters **10**, pp 243–252, 1999.
87. 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.
88. 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.
89. 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.
90. 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.
91. 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.
92. T. Hastie, R. Tibshirani, Non-parametric logistic and proportional odds regression, JRSS-C (Applied Statistics) **36**, pp. 260–276, 1987.
93. M. Dash, H. Liu, P. Scheuermann, K. L. Tan, Fast hierarchical clustering and its validation, Data & Knowledge Engineering **44**, pp 109–138, 2003.
94. P. Cortez, A. M. Gonçalves Silva, Using data mining to predict secondary school student performance, In Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008) (pp. 5–12). EUROSIS-ETI, 2008.
95. I-Cheng Yeh, King-Jang Yang, Tao-Ming Ting, Knowledge discovery on RFM model using Bernoulli sequence, Expert Systems with Applications **36**, pp. 5866-5871, 2009.
96. 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.
97. 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.
98. P. Zhong, M. Fukushima, Regularized nonsmooth Newton method for multi-class support vector machines, Optimization Methods and Software **22**, pp. 225-236, 2007.
99. 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.

100. M. Koivisto, K. Sood, Exact Bayesian Structure Discovery in Bayesian Networks, The Journal of Machine Learning Research **5**, pp. 549–573, 2004.
101. Nash, W.J.; Sellers, T.L.; Talbot, S.R.; Cawthor, A.J.; Ford, W.B. 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 48; Sea Fisheries Division, Department of Primary Industry and Fisheries: Orange, NSW, Australia, 1994.
102. T.F. Brooks, D.S. Pope, A.M. Marcolini, Airfoil self-noise and prediction. Technical report, NASA RP-1218, July 1989.
103. J.S. Simonoff, Smooting Methods in Statistics, Springer - Verlag, 1996.
104. I.Cheng Yeh, Modeling of strength of high performance concrete using artificial neural networks, Cement and Concrete Research. **28**, pp. 1797-1808, 1998.
105. D. Harrison and D.L. Rubinfeld, Hedonic prices and the demand for clean ai, J. Environ. Economics & Management **5**, pp. 81-102, 1978.
106. 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.
107. D.C. Liu, J. Nocedal, On the Limited Memory Method for Large Scale Optimization, Mathematical Programming B. **45**, pp. 503–528, 1989.
108. Gropp, W.; Lusk, E.; Doss, N.; Skjellum, A. A high-performance, portable implementation of the MPI message passing interface standard. Parallel Comput. 1996, 22, 789–828.
109. Chandra, R. Parallel Programming in OpenMP; Morgan Kaufmann: Cambridge, MA, USA, 2001.