MDPI

Article

Application of feature construction techniques in forest fire duration prediction data

Constantina Kopitsa[1], Ioannis G. Tsoulos[2]*, Andreas Miltiadous[3], Vasileios Charilogis[4]

[1] Department of Informatics and Telecommunications, University of Ioannina, Greece;k.kopitsa@uoi.gr
[2] Department of Informatics and Telecommunications, University of Ioannina, Greece; itsoulos@uoi.gr
[3] Department of Informatics and Telecommunications, University of Ioannina, Greece; a.miltiadous@uoi.gr
[4] Department of Informatics and Telecommunications, University of Ioannina, Greece; v.charilog@uoi.gr
* Correspondence: itsoulos@uoi.gr

Abstract: From early times, humanity has expressed a profound need to predict the future. In ancient Greece, oracles held a place of high respect, and influence. In modern times, the quest for accurate forecasting, has shifted to the realms, of research and science**.** empowered by advanced computational tools, provided by Artificial Intelligence, particularly Machine Learning. One of the fields, where Machine Learning, has established a fertile ground is in the domain, of forest fire management. Forest fires, pose a major threat to both human, and animal life, with significant economic and social impacts. A reliable prediction system is crucial for mitigating these effects, especially during summer months, dry seasons, and in high-risk areas, such as the Mediterranean. This study, explores feature construction, and selection methods, applied to forest fire data, collected over 10 years in Greece, incorporating prevailing weather conditions at ignition, and during suppression. By applying techniques like Principal Component Analysis (PCA), Minimum Redundancy Maximum Relevance (MRMR) feature selection, and Grammatical Evolution for feature construction, this research aims to identify key factors influencing fire duration. These techniques, have become invaluable allies, in addressing complex predictive challenges, and advancing, our understanding of future events. Our approach, leverages advanced computational methods, to analyze complex datasets, providing a deeper understanding of the primary drivers, of wildfire behavior, and enabling more effective mitigation strategies.

Keywords: Forest fires; Machine learning; Neural networks; Feature Construction; Genetic Programming; Grammatical Evolution

Citation: Kopitsa, C.; Tsoulos, I.G.;Miltiadous, A.;Charilogis V. Application of feature construction techniques in forest fire duration prediction data. *Journal Not Specified* **2024**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

1. Introduction

2. Materials and Methods

What to show here: full description of the used datasets and a brief description of the used methods and references to related works.

2.1. The used dataset

2.2. The used methods

2.2.1. The PCA method

The Principal Component Analysis (PCA) technique, introduced by mathematician Karl Pearson in 1901 [1], and developed by Harold Hotelling (1933). This technique, operates on the principle when data from a higher-dimensional space is transformed into a lower-dimensional space, the resulting lower-dimensional representation should retain the maximum variance, of the original data.

Notably, it is worth mentioning that the use of PCA, on larger datasets, became practical only after the advent of electronic computers, which made it computationally feasible, to handle datasets, beyond trivial sizes [2]. Continuing, with the applications of PCA, it is a widely utilized technique in exploratory data analysis, and machine learning,

particularly, in building predictive models. It is an unsupervised learning method, designed to analyze, the relationships among a set of variables. Often referred to as a form of general factor analysis, it involves regression to determine a line of best fit. The primary objective, of PCA, is to reduce the dimensionality of a dataset, while retaining the most significant patterns, and relationships, among the variables, all without requiring prior knowledge, of the target variables [3].

Next, we will briefly reference studies, that have utilized PCA, covering different areas, such as: statistical physics, genetic improvement, face recognition, economic & environmental sciences, medical prediction, e.t.c. Explicitly, the research conducted, by Park [4], highlights the reasons behind the success of the PCA technique, for lattice systems. The study's primary limitation lies in the dependency of the proposed formula's accuracy on the dataset size. Specifically, the results achieve full precision, only under the condition of an infinite dataset. This constraint restricts the practical applicability, of the method, when working with finite or limited data, a common scenario, in real-world analyses.

Additionally, the work of Sarma et al. [5] utilizes PCA, in order to evaluate, morphometric traits, under a multivariate approach. The findings suggest, that PCA could significantly enhance the genetic improvement. Noteworthy, is the fact, that the 64.29%, of the total variance explained, can be considered relatively low. This suggests, that a significant amount of unexplained information remains, which is not captured, by the four principal components. Moreover, Gambardella et al. used the PCA technique for monitoring the cultivation, of cannabis, in Albania. Specifically, with PCA they remove redundant spectral information from multiband datasets [6]. The article, by Slavkovic and Jevtic [7], presents the implementation of a face recognition system based on the Principal Component Analysis (PCA) algorithm. The PCA technique was utilized by Hargreaves [8], for stock selection, specifically to identify, a limited number of stock variables, that could effectively aid, in determining winning stocks.

Moreover, Xu et al. presents an interesting example of a modified application of Principal Component Analysis (PCA), utilizing, both linear and non-linear methods, through Kernel PCA (KPCA), in combination, with the Adaptive Boosting (AdaBoost) algorithm [9]. In the study of Zhang [10], a neural network model, combining PCA and Levenberg-Marquardt [11] was developed, to efficiently, and accurately analyze, and predict the interaction between IAQ and its influencing factors. In particular, it was examined indoor air quality (IAQ), and its relationship, with building features, and environmental conditions.

In the work of Akinnuwesi et al [12], a hybrid approach was suggested combining Principal Component Analysis (PCA), and Support Vector Machine (SVM) [13]. They create, the Breast Cancer Risk Assessment and Early Diagnosis (BC-RAED) model, designed to accurately detect BCa, in its early stages. PCA, was initially applied to extract features, during the first preprocessing stage, followed by further feature reduction, in the second stage. The multi-preprocessed data, were analyzed for breast cancer risk, and diagnosis using SVM. The BC-RAED model, achieved, an accuracy of 97.62%, a sensitivity of 95.24%, and a specificity of 100% in assessing, and diagnosing breast cancer risk.

Subsequently, we will briefly mention certain studies, that have been conducted, in the field of Forest Fires. Guan's research focuses on forest fire prediction using PCA-preprocessed data. The preprocessing step removed irrelevant information, simplifying analysis. Linear regression and random forest methods were then applied, revealing temperature, relative humidity, wind, and rain as the most influential factors in forest fire occurrence [14].

A novel model was developed, by Nikolov, using meteorological forecast data as input. Principal Component Analysis (PCA), with orthogonal rotation, was applied to reduce 195 meteorological variables, from the NARR dataset, to a smaller set of significant fire-ignition predictors, later used in logistic regression, to calculate wildfire ignition probabilities [15]. Also, a recent publication focuses on predicting wildfire ignitions, caused by lightning strikes, which account for the largest area burned annually, in the extratropical Northern

Hemisphere. Principal Component Analysis (PCA), played a key role, in reducing 611 potential predictors, to 13 principal components, which were used in logistic regression to identify the primary factors influencing lightning occurrence [16].

2.2.2. The MRMR method

The min-redundancy max-relevance (MRMR) algorithm, introduced by Chris Ding and Hanchuan Peng [17]. This method aims to optimize feature selection, by minimizing redundancy, and maximizing relevance [18]. In sum, MRMR enhances relevance-only methods, such as using an f-test between the target, and the features. When two features are similar, MRMR prioritizes only the one, with the highest relevance.

The study, by Zhao, extends traditional MRMR methods, by introducing, a non-linear feature redundancy measure, and a model-based feature relevance measure, which are tested, on synthetic, and real-world datasets. Based on its empirical success, MRMR is integrated, into Uber's marketing machine learning platform, to automate, the creation and deployment, of scalable targeting, and personalization models [19].

Moreover, Wu et al. proposed that the MRMR algorithm is utilized in conjunction with a Random Forest model [20] to perform feature selection in the context of air quality prediction. MRMR, is employed, to determine which variables have the most significant impact, on the air quality index (AQI), while minimizing redundancy among them [21].

The article, by Elbeltagi, is an innovative approach, for estimating maize chlorophyll, by integrating hyperspectral indices with cutting-edge, six advanced machine learning techniques. The MRMR algorithm was incorporated into the process to enhance feature selection by pinpointing the most significant spectral bands, minimizing data redundancy, and boosting model efficiency [22].

In the energy sector, Liu conducted the following research, offering an improved method, for predicting transient stability, in power systems. The MRMR algorithm, is applied for feature selection, with minimal redundancy and maximum relevance, providing an enhanced approach, for forecasting transient stability, in power systems. This approach addresses, the limitations of previous methods, such as low accuracy, difficult applicability, and high computational cost, while incorporating the "winner take all" (WTA) technique, for ensemble learning, and enhanced precision [23].

Eristi, also refers to the energy sector. Specifically, this paper presents, a new PD detection system, that combines spectral analysis, spectrogram analysis, deep learning algorithms, MRMR and ensemble machine learning (EML) [24]. The most impactful features, are identified, by performing MRMR feature selection analysis, on the extracted deep features [25].

Zhang, employed an Acoustic Emission (AE) technique to monitor, inaccessible areas of large storage tank floors, utilizing AE sensors positioned externally to the tank. The implemented algorithm, effectively distinguishes corrosion signals from interference signals, particularly drop-back signals induced by condensation. Experimental studies were conducted both in laboratory settings and in field environments, focusing on Q235 steel. Seven characteristic AE features, derived from signal hits, and frequency, were extracted, and subsequently selected for pattern recognition, using the MRMR method [26].

Additionally, Karamouz et al., proposed a methodology to examine the effects of climate change, on sea level variations, in coastal areas, using an artificial neural network model. Feature selection techniques, including MRMR and Mutual Information (MI), are employed to identify the most suitable predictors for the neural network input [27].

2.2.3. The Neural Network Construction method

Another machine learning method introduced recently that is based on Grammatical Evolution [28] is the construction of artificial neural networks [29]. In this work the architecture of the neural network is produced through a series of generations of the underlying genetic algorithm by reducing the training error of the neural network. Furthermore, the method is able to identify the best set of parameters for the neural network. This method

can also retain only a small portion of features from the original objective problem, significantly reducing the information required to reduce the training error. This method was used in a series of practical problems, such as the identification of amide I bonds [30], solution of differential equations [31], incorporation in the detection of Parkinson's disease [32], usage in the estimation of performance of students [33], autism screening [34] etc.

The used grammar for the construction of neural networks expressed in Backus–Naur (BNF) form [35] is shown in Figure 1. Numbers in parentheses represent the sequence number of each production rule. The constant n stands for the number of input features.

```
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 incorporated in the construction of neural networks.

This grammar produces artificial neural networks in the following form:

$$\mathbf{NN}(\overrightarrow{x}, \overrightarrow{w}) = \sum_{i=1}^{H} w_{(n+2)i-(n+1)} \sigma \left(\sum_{j=1}^{n} x_j w_{(n+2)i-(n+1)+j} + w_{(n+2)i} \right) \quad (1)$$

The symbol H defines the number of processing nodes (weights). The sigmoid function $\sigma(x)$ is used as the activation function of neural network and it is defined as:

$$\sigma(x) = \frac{1}{1+\exp(-x)} \quad (2)$$

The main steps of the algorithm are:

1. **Initialization step**.
   (a) **Set** the number of used chromosomes N_c. Each chromosome is a set of randomly selected integers. These integer values represent rule number in the extended BNF grammar previously presented.
   (b) **Set** the maximum number of allowed generations N_g.
   (c) **Set** the selection rate $p_s \in [0,1]$ and the mutation rate $p_m \in [0,1]$.
   (d) **Set** $k = 0$, the generation number.
2. **Fitness calculation step**.
   (a) **For** each chromosome g_i, $i = 1, \ldots, N_c$ **do**
      i. **Create** using the grammar of figure 1 the corresponding neural network $\mathrm{NN}_i(\overrightarrow{x}, \overrightarrow{w})$
      ii. **Set** as $f_i = \sum_{j=1}^{M} \left(\mathrm{NN}_i(\overrightarrow{x}_j, \overrightarrow{w_i}) - y_j\right)^2$ the fitness of chromosome i. The set $(\overrightarrow{x_j}, y_j)$, $j = 1, ..., M$ stands for the train set of the objective problem.

(b) **End For**

3. **Genetic operations step**.
   (a) Application of Selection operator. The chromosomes of the population are sorted according to their fitness values and the best $(1 - p_s) \times N_c$ chromosomes are copied to the next generation. The remaining are replaced by new chromosomes produced during crossover and mutation.
   (b) Application of Crossover operator. In this step $p_s \times N_c$ new chromosomes will be created from the original ones. For each set c_1, c_2 of new chromosomes that will be created, two chromosomes g_a and g_b are selected from the old population using tournament selection. The new chromosomes are created using one - point crossover between g_a and g_b. An example of this operation is shown graphically in Figure 2.
   (c) Application of Mutation operator. For each element of every chromosome a random number $r \in [0, 1]$ is selected. The corresponding element is changed randomly when $r \leq p_m$.
4. **Termination check step**.
   (a) **Set** $k = k + 1$
   (b) **If** $k \leq N_g$ then goto Fitness Calculation Step.
5. **Application to the test set**.
   (a) **Obtain** the best chromosome g^* from the genetic population.
   (b) **Create** the corresponding neural network $\mathrm{NN}^*\left(\overrightarrow{x}, \overrightarrow{w}\right)$
   (c) **Apply** this neural network to the test set of the objective problem and report the corresponding error (test error).

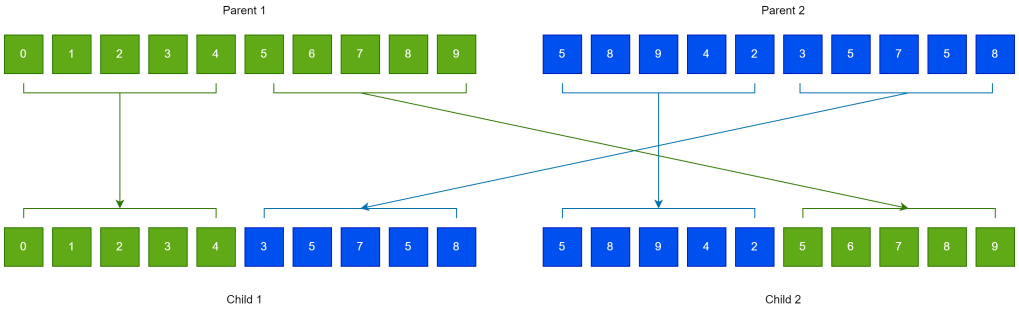

Figure 2. An example of the one - point crossover operation used in Grammatical Evolution.

2.2.4. The Feature Construction method

Another approach discussed here and used in the conducted experiments is the feature construction technique initially proposed in [36]. This approach creates artificial features from the original ones using the Grammatical Evolution procedure. The new features are non - linear mappings of the original ones. This method has been used in a series of practical cases during the past years, such as Spam Identification [37], Fetal heart classification [38], EEG signal processing [39,40] etc. The extended version of BNF grammar used during the feature construction process is outlined in Figure 3.

Figure 3. The extended BNF grammar used in the feature construction process.

```
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-1)
<digitlist>::=<digit>                  (0)
          | <digit><digit>             (1)
          | <digit><digit><digit>      (2)
<digit>  ::= 0 (0)
          | 1 (1)
          .......
          | 9 (9)
```

The main steps of the used algorithm have as follows:

1. **Initialization step**.
   (a) **Define** the number of used chromosomes N_c.
   (b) **Define** the maximum number of allowed generations N_g.
   (c) **Set** the selection rate $p_s \in [0,1]$ and the mutation rate $p_m \in [0,1]$.
   (d) **Set** as N_f the number of desired features that will be created.
   (e) **Set** $k = 0$, the generation number.
2. **Fitness calculation step**.
3. **For** $i = 1, \ldots, N_c$ **do**
   (a) **Create** with the assistance of Grammatical Evolution, a set of N_f artificial features from the original ones, for chromosome g_i.
   (b) **Transform** the original train set using the previously produced features. Represent the new set as $\text{TR} = \left(x_{g_i,j}, t_j\right),\ j = 1, ..M$
   (c) **Apply** a machine learning model denoted as C on set TR and train this model and denote as $C(x)$ the output of this model for any input pattern x.
   (d) **Calculate** the fitness f_i as:

$$f_i = \sum_{j=1}^{M} \left(C\left(x_{g_i,j}\right) - t_j\right)^2 \tag{3}$$

   The Radial Basis Function (RBF) networks [41,42] were used as the machine learning models $C(x)$ in the current work. This machine learning model was chosen because of the significantly shorter training time it requires compared to other machine learning models.
4. **End For**

5. **Genetic operations step.** Perform the same genetic operators as in the case of construction neural networks, discussed previously.
6. **Termination check step**.
   (a) **Set** $k = k + 1$
   (b) **If** $k \leq N_g$ then goto Fitness Calculation Step.
7. **Application to the test set**.
   (a) **Obtain** the chromosome g^* with the lowest fitness value.
   (b) **Create** the N_f artificial features that correspond to this chromosome
   (c) **Apply** the N_f features to the train set and produce the mapped training set $\text{TR} = \left(x_{g_i,j}, t_j\right),\ j = 1, ..M$
   (d) **Train** a machine learning model on the produced training set. An artificial neural network [43,44] with $H = 10$ processing nodes is used in the current work. This neural network was trained using a BFGS variant of Powell [45].
   (e) **Apply** the new features to the test set of the objective problem and create the set $\text{TT}=\left(x_{g_i,j}, t_j\right),\ j = 1, ..K$
   (f) **Apply** the machine learning model on set TT and report the test error.

3. Results

The experiments were conducted using the freely available programming tool of WEKA [46]. The software, which is written in the JAVA programming language to be portable, can be downloaded freely from https://ml.cms.waikato.ac.nz/weka/(accessed on 14 September 2024) or it can be found in the repositories of most Linux systems. The WEKA software is a collection of machine learning and data analysis tools and it contains also some visualization tools for modeling. The WEKA has been used with success in many cases, such as educational problems [47,48], medical problems [49,50] etc. 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 experimental results using the methods mentioned in the previous section and the 10 modified datasets from the Hellenic Fire Service are listed in Table 1. The following applies to the tables of experimental results:

1. The column YEAR denotes the year of recording.
2. The column BAYES the application of the Naive Bayes [51] method to the corresponding dataset.
3. The column ADAM represents the usage of the ADAM optimizer [52] for the training of a neural network with $H = 10$ processing nodes.
4. The column BFGS denotes the incorporation of the BFGS optimizer [45] to train a neural network with $H = 10$ processing nodes.
5. The column MRMR denotes the results obtained by the application of a neural network trained with the BFGS optimizer on two features selected using the MRMR technique.
6. The column PCA stands for the results obtained by the application of a neural network trained with the BFGS optimizer on two features created using the PCA technique. The PCA variant implemented in MLPACK software [53] was incorporated to create these features.
7. The column NNC denotes the usage of the method of Neural Network Construction on the proposed datasets. The software that implements this method was obtained from [54].
8. The column FC represents the usage of the previously mentioned method for constructing artificial features. For the purposes of this article two artificial features were created. These features were produced and evaluated using the QFc software [55].

Table 1. Experimental results using a series of machine learning methods for the prediction of forest fire duration.

YEAR	BAYES	ADAM	BFGS	MRMR	PCA	NNC	FC
2014	11.41%	13.00%	12.38%	9.68%	15.50%	9.21%	8.04%
2015	10.49%	11.94%	11.25%	8.49%	15.03%	9.17%	7.51%
2016	10.79%	12.95%	11.88%	9.45%	12.93%	10.12%	8.60%
2017	53.36%	12.68%	12.65%	12.65%	12.64%	12.61%	12.66%
2018	9.39%	10.48%	14.97%	9.21%	10.49%	9.29%	7.72%
2019	7.79%	9.44%	9.66%	8.39%	9.72%	7.03%	6.62%
2020	40.26%	9.56%	9.80%	9.55%	9.76%	9.50%	9.61%
2021	11.81%	11.06%	12.90%	10.57%	11.03%	10.80%	9.55%
AVERAGE	**19.41%**	**11.39%**	**11.94%**	**9.75%**	**12.14%**	**9.72%**	**8.79%**

In Figure 4, the results of the Wilcoxon test for pairwise comparisons of the classification models are presented, providing valuable insights into the statistical significance of their performance differences. Below is a more in-depth analysis, focusing on the interpretation of results and a deeper understanding of the relationships between the models. The overall result of the Wilcoxon test ($p < 0.5$) confirms that statistically significant differences exist among the models' performances. This indicates that the models are not equivalent in terms of their accuracy, with some clearly outperforming others. The comparison between FC and NNC ($p = 0.039$) reveals a statistically significant difference, though the p-value is relatively close to the significance threshold (commonly $p < 0.05$). This suggests that while FC outperforms NNC, the difference is not exceedingly pronounced. This outcome might be influenced by specific data characteristics or variations in the models' stability. The comparison between FC and PCA ($p = 0.016$) shows a clearer statistically significant difference. FC's performance is evidently superior to PCA's, which may be attributed to the fundamental differences in their methodologies. PCA, as a dimensionality reduction technique, might lose critical information in the data, leading to lower accuracy in certain scenarios. For the comparison between FC and MRMR ($p = 0.039$), a statistically significant difference is observed once again. Similar to the FC-NNC comparison, this difference exists but is not highly pronounced. MRMR, which selects features based on mutual information, might not perform as consistently across all datasets, giving FC an edge. The comparison between FC and BFGS ($p = 0.016$) indicates an even more distinct difference. BFGS, as an optimization method, may lack the flexibility or precision required for classification tasks, allowing FC to demonstrate more stable and superior performance in this case. The difference between FC and ADAM ($p = 0.023$) is also statistically significant. While ADAM is generally considered an effective algorithm in many contexts, FC appears to outperform it in this analysis, possibly due to better adaptability to the specific characteristics of the dataset. The most striking difference is seen in the comparison between FC and BAYES ($p = 0.0078$). The very low p-value strongly confirms a statistically significant difference, highlighting FC's clear superiority over BAYES. Notably, BAYES has exhibited high variability in its performance, especially in 2017, when it recorded a very high error rate. This variability likely reduces its overall reliability, which is reflected in this comparison. In summary, FC consistently outperforms all other models in this analysis, with statistically significant differences observed in all pairwise comparisons. The differences are not only numerical but also conceptual, as FC's superior performance can likely be attributed to its stability, flexibility, and ability to handle the data's nuances more effectively. In contrast, models like BAYES and PCA seem more affected by changes in the data or problem conditions. This analysis highlights FC as the most reliable and high-performing model among those evaluated.

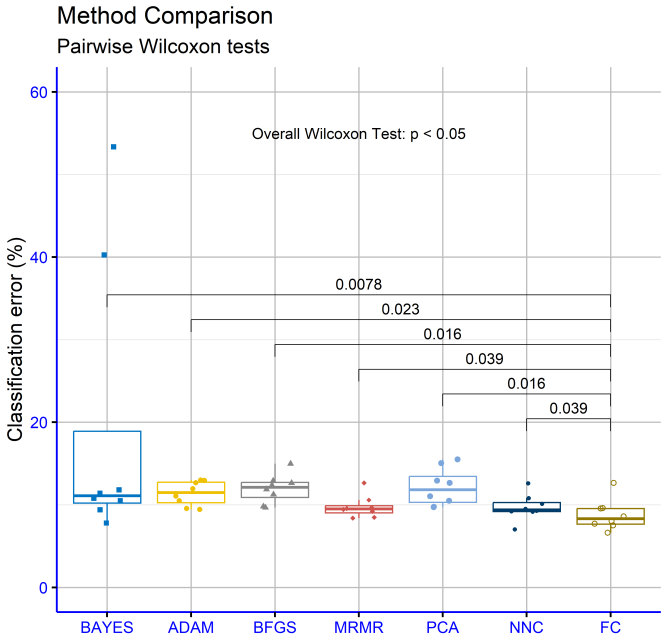

Figure 4. Statistical comparison of the used machine learning techniques.

4. Conclusions

Author Contributions: C.K., V.C. and I.G.T. conceived of the idea and the methodology, and C.K. and V.C. implemented the corresponding software. C.K. conducted the experiments, employing objective functions as test cases, and provided the comparative experiments. A.S. 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.

Data Availability 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. A. Maćkiewicz, W. Ratajczak, Principal components analysis (PCA). Computers & Geosciences **19**, pp. 303-342, 1993.
2. J. Cadima, I.T. Jolliffe, Principal component analysis: a review and recent developments, National Library of Medicine. 2016. Available online: https://pmc.ncbi.nlm.nih.gov/articles/PMC4792409/ (accessed on 16 November 2024).
3. i2tutorials. What are the Pros and Cons of the PCA? October 1st, 2019. Available online: https://www.i2tutorials.com/what-are-the-pros-and-cons-of-the-pca/ (accessed on 16 November 2024).
4. S.C. Park, Physical Meaning of Principal Component Analysis for Lattice Systems with Translational Invariance. arXiv preprint arXiv:2410.22682, 2024.
5. O. Sarma, M.A. Rather, S. Shahnaz, R.S. Barwal, Principal Component Analysis of Morphometric Traits in Kashmir Merino Sheep. Journal of Advances in Biology & Biotechnology **27**, pp. 362-69, 2024.
6. C. Gambardella, R. Parente, A. Ciambrone, M. Casbarra, A Principal Components Analysis – Based Method for the Detection of Cannabis Plants Using Representation data by Remote Sensing, Knowledge Extractions from data Using Machine Learning, volume 6 (10), 2021.
7. M. Slavkovic, D. Jevtic, Face Recognition Using Eigenface Approach, Serbian Journal of Electrical Engineering **9**, pp. 121-130, 2012.

8. C.A. Hargreaves, C.K. Mani, The Selection of Winning Stocks Using Principal Component Analysis, American Journal of Marketing Research **1**, pp. 183 – 188, 2015.
9. Z. Xu, F. Guo, H. Ma, X. Liu, L. Gao, On Optimizing Hyperspectral Inversion of Soil Copper Content by Kernel Principal Component Analysis. Remote Sensing **16**, 2024.
10. H. Zhang, R. Srinivasa, X. Yang, S. Ahrentzen, E.S. Coker, A. Alwisy, Factors influencing indoor air pollution in buildings using PCA – LMBP neural network: A case study of a university campus, Building and environment **225**, 2022.
11. M.I. Lourakis, A brief description of the Levenberg-Marquardt algorithm implemented by levmar, Foundation of Research and Technology **4**, pp. 1-6, 2005.
12. B.A. Akinnuwesi, B.O. Macaulay, B.S. Aribisala, Breast cancer risk assessment and early diagnosis using Principal Component Analysis and support vector machine techniques, Informatics in medicine unlocked **21**, 100459, 2020.
13. M. Awad, R. Khanna, M. Awad, R. Khanna, Support vector machines for classification. Efficient learning machines: Theories, concepts, and applications for engineers and system designers, pp. 39-66, 2015.
14. R. Guan, Predicting forest fire with linear regression and random forest, Highlights in Science, Engineering and Technology **44**, pp. 1-7, 2023.
15. N. Nikolov, P. Bothwell, J. Snook, Developing a gridded model for probabilistic forecasting of wildland-fire ignitions across the lower 48 States. USFS-CSU Joint Venture Agreement Phase 2 (2019-2021)-Final Report. Fort Collins, CO: US Department of Agriculture, Forest Service, Rocky Mountain Research Station. 33p, 2022.
16. N. Nikolov, P. Bothwell, J. Snook, Probalistic forecasting of lightning strikes over the Continental USA and Alaska: Model development and verification. Scientific Journal, Fire (7). Available from: https://research.fs.usda.gov/treesearch/68447 (accessed 20 November 2024).
17. C. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data. Journal of bioinformatics and computational biology **3**, pp. 185-205, 2005.
18. S. Ramirez – Gallego, I. Lastra, D. Martinez – Rego, V. Bolon – Canedo, J.M. Benitez, F. Herrera, A. Alonso – Betanzos, Fast-mRMR: Fast Minimum Redundancy Maximum Relevance Algorithm for High – Dimensional Big Data, International Journal of Intelligent Systems **32**, pp. 134 – 152, 2017.
19. Zhao, Z., Anand, R., & Wang, M. (2019, October). Maximum relevance and minimum redundancy feature selection methods for a marketing machine learning platform. In 2019 IEEE international conference on data science and advanced analytics (DSAA) (pp. 442-452). IEEE.
20. S.J. Rigatti, Random forest. Journal of Insurance Medicine **47**, pp. 31-39, 2017.
21. Wu, H., Yang, T., Li, H., & Zhou, Z. (2023). Air quality prediction model based on mRMR–RF feature selection and ISSA–LSTM. Scientific Reports, 13(1), 12825.
22. Elbeltagi, A., Nagy, A., Szabo, A., Nxumalo, G.S., Bodi, E.B., Tamas, J. Hyperspectral indices data fusion-based machine learning enhanced by MRMR algorithm for estimating maize chlorophyll content. Frontiers in Plant Science, Section Technical Advances in Plant Science, volume 15, October 2024.
23. Liu, J., Sun, H., Li, Y., Fang, W., & Niu, S. (2020). An improved power system transient stability prediction model based on mRMR feature selection and WTA ensemble learning. Applied Sciences, 10(7), 2255.
24. Dietterich, T. G. (2000, June). Ensemble methods in machine learning. In International workshop on multiple classifier systems (pp. 1-15). Berlin, Heidelberg: Springer Berlin Heidelberg.
25. Eristi, B. (2024). A New Approach based on Deep Features of Convolutional Neural Networks for Partial Discharge Detection in Power Systems. IEEE Access.
26. Li, Y., Zhang, Y., Zhu, H., Yan, R., Liu, Y., Sun, L., & Zeng, Z. (2015). Recognition algorithm of acoustic emission signals based on conditional random field model in storage tank floor inspection using inner detector. Shock and Vibration, 2015(1), 173470.
27. Karamouz, M., Zahmatkesh, Z., Nazif, S., & Razmi, A. (2014). An evaluation of climate change impacts on extreme sea level variability: Coastal area of New York City. Water resources management, 28, 3697-3714.
28. M. O'Neill, C. Ryan, Grammatical evolution, IEEE Trans. Evol. Comput. **5,**pp. 349–358, 2001.
29. I.G. Tsoulos, D. Gavrilis, E. Glavas, Neural network construction and training using grammatical evolution, Neurocomputing **72**, pp. 269-277, 2008.
30. 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.
31. I.G. Tsoulos, D. Gavrilis, E. Glavas, Solving differential equations with constructed neural networks, Neurocomputing **72**, pp. 2385-2391, 2009.
32. 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.
33. 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.
34. E.I. Toki, J. Pange, G. Tatsis, K. Plachouras, I.G. Tsoulos, Utilizing Constructed Neural Networks for Autism Screening, Applied Sciences **14**, 3053, 2024.

35. 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.
36. Dimitris Gavrilis, Ioannis G. Tsoulos, Evangelos Dermatas, Selecting and constructing features using grammatical evolution, Pattern Recognition Letters **29**,pp. 1358-1365, 2008.
37. Dimitris Gavrilis, Ioannis G. Tsoulos, Evangelos Dermatas, Neural Recognition and Genetic Features Selection for Robust Detection of E-Mail Spam, Advances in Artificial Intelligence Volume 3955 of the series Lecture Notes in Computer Science pp 498-501, 2006.
38. George Georgoulas, Dimitris Gavrilis, Ioannis G. Tsoulos, Chrysostomos Stylios, João Bernardes, Peter P. Groumpos, Novel approach for fetal heart rate classification introducing grammatical evolution, Biomedical Signal Processing and Control **2**,pp. 69-79, 2007
39. Otis Smart, Ioannis G. Tsoulos, Dimitris Gavrilis, George Georgoulas, Grammatical evolution for features of epileptic oscillations in clinical intracranial electroencephalograms, Expert Systems with Applications **38**, pp. 9991-9999, 2011
40. A. T. Tzallas, I. Tsoulos, M. G. Tsipouras, N. Giannakeas, I. Androulidakis and E. Zaitseva, Classification of EEG signals using feature creation produced by grammatical evolution, In: 24th Telecommunications Forum (TELFOR), pp. 1-4, 2016.
41. J. Park and I. W. Sandberg, Universal Approximation Using Radial-Basis-Function Networks, Neural Computation 3, pp. 246-257, 1991.
42. 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.
43. C. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
44. G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of Control Signals and Systems **2**, pp. 303-314, 1989.
45. M.J.D Powell, A Tolerant Algorithm for Linearly Constrained Optimization Calculations, Mathematical Programming **45**, pp. 547-566, 1989.
46. M. Hall, F. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update. ACM SIGKDD explorations newsletter **11**, pp. 10-18, 2009.
47. S.B. Aher, L.M.R.J. Lobo, Data mining in educational system using weka. In International conference on emerging technology trends, Foundation of Computer Science **3**, pp. 20-25, 2011.
48. S. Hussain, N.A. Dahan, F.M. Ba-Alwib, N. Ribata, Educational data mining and analysis of students' academic performance using WEKA. Indonesian Journal of Electrical Engineering and Computer Science **9**, pp. 447-459, 2018.
49. A.K. Sigurdardottir, H. Jonsdottir, R. Benediktsson, Outcomes of educational interventions in type 2 diabetes: WEKA data-mining analysis. Patient education and counseling **67**, pp. 21-31, 2007.
50. M.N. Amin, A. Habib, Comparison of different classification techniques using WEKA for hematological data. American Journal of Engineering Research **4**, pp. 55-61, 2015.
51. G.I. Webb, E. Keogh, R. Miikkulainen, Naïve Bayes, Encyclopedia of machine learning **15**, pp. 713-714, 2010.
52. 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.
53. Ryan R. Curtin, James R. Cline, N. P. Slagle, William B. March, Parikshit Ram, Nishant A. Mehta, Alexander G. Gray, MLPACK: A Scalable C++ Machine Learning Library, Journal of Machine Learning Research **14**, pp. 801-805, 2013.
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. I.G. Tsoulos, QFC: A Parallel Software Tool for Feature Construction, Based on Grammatical Evolution. Algorithms **15**, 295, 2022.