Genetic Programming-based AutoML for EEG Signal Classification - A Comparative Study

I. M. Miranda[1], C. Aranha[2], A. P. L. de Carvalho[3], L. P. F. Garcia[1]

[1] Universidade de Brasília, Brazil
icaro.miranda@aluno.unb.br, luis.garcia@unb.br
[2] Tsukuba University, Japan
caranha@cs.tsukuba.ac.jp
[3] Universidade de São Paulo, Brazil
andre@icmc.usp.br

Abstract. End-to-end Machine Learning (ML) applications using complex data often need to investigate several alternatives for the data modeling pipeline before a good solution is found. This process, which is time-consuming and subjective, can benefit from an automated solution design by using Automated Machine Learning (AutoML). End-to-end AutoML allows automated data preparation, modeling, and evaluation of ML pipelines, increasing the chances of arriving at a good solution. AutoML can implement this optimization with different strategies. Among them, Genetic Programming (GP) stands out for its ability to create pipelines of arbitrary format, allowing high interpretability and the customization of information from the data context. This paper proposes and compares two approaches of end-to-end AutoML optimized with GP for a time series classification problem, the classification of Electroencephalogram (EEG) signals. We selected this dataset because of the signals' high complexity, spatial and temporal co-variance, and non-stationarity. For the AutoML experiments, four different domain-based data characterization measures are evaluated. The analysis of the data characterization measures shows that using only spectral or time-domain features does not lead to pipelines with good predictive performance. Our experimental results also reveal how AutoML can generate more accurate and interpretable solutions than the literature's complex and *ad hoc* models. The proposed approach makes it easier to analyze dimensional reduction through fitness convergence, tree depth, and extracted features.

Categories and Subject Descriptors: I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems; I.2.6 [**Artificial Intelligence**]: Learning

Keywords: AutoML, Classification, EEG, End-to-end Machine Learning, Genetic Programming, Sleep Spindles

1. INTRODUCTION

End-to-end Machine Learning (ML) experiments for real-world applications require a pipeline of steps, including an understanding of the application context, data exploration, data transformation and preprocessing, data modeling, model evaluation, and solution implementation [Azevedo and Santos 2008]. Each of these steps, implemented by a flow of techniques and algorithms, can be subjective and time-consuming. However, a significant part of this process can be automated using Automated Machine Learning (AutoML) [Zöller and Huber 2021] [Hutter et al. 2019].

The large number of available ML algorithms and data preprocessing techniques that can be selected, together with the complexity of the application, make the design of a promising pipeline a challenging task. A data scientist can manually define a sequence of operations for simple applications. Nevertheless, in most situations, it is not easy to know it *a priori*. Therefore, the search and optimization techniques can be employed to automatically build an adequate pipeline based on the data [Zöller and Huber 2021]. This data-driven automation is called AutoML [Hutter et al. 2019].

Popular AutoML open-source frameworks, such as *Auto-sklearn* [Hutter et al. 2019], use Bayesian optimization to find good pipelines in a search space by selecting and optimizing preprocessing operations and performing hyper-parameter tuning on ML algorithms. Other frameworks use different approaches, e.g., H2O [LeDell and Poirier 2020] employs random search and stacking of the best pipelines found, and TPOT (Tree-based Pipeline Optimization Tool) [Olson and Moore 2019] uses Genetic Programming (GP) to build them.

GP's main advantages are easy parallelization, high customization, interpretability, control of computational resources, and insertion of application domain information [Zöller and Huber 2021]. It allows the creation of expressions or computer programs from combinations of user-defined operators. Its flexibility makes it easier to solve problems in different contexts with different data types [Poli et al. 2008], even time series analysis [Miranda et al. 2019].

A time series is a data type where observations show a time ordering - and they depend on each other based on this ordering [Bontempi et al. 2012]. Regarding the use of traditional ML algorithms on these data, a step of transforming the temporal features through descriptive measures is necessary to represent the problem most properly [Motamedi-Fakhr et al. 2014].

In the context of bio-signals, e.g., Electroencephalogram (EEG) signals are particularly problematic time series due to their high complexity, spatial and temporal co-variance, and non-stationarity [Kevric and Subasi 2017]. The classification of EEG signals is usually performed by a medical specialist, which can take a long time, be tiring, error-prone, and introduce biases or be affected by mood, distraction, and years of experience [Bontempi et al. 2012][Motamedi-Fakhr et al. 2014].

This paper compares state-of-the-art AutoML tools with the GP-based AutoML for identifying events in EEG signals, which is considered a complex benchmark [Motamedi-Fakhr et al. 2014]. Besides, we propose a characterization of EEG data using four feature groups: Statistical, Spectral, Complexity, and Time Series bases. For such, we use the Sleep Spindles public database of EEG signals [Devuyst et al. 2011] to analyze: the quality of pipelines found for classification metrics, the selected features, and the pipeline structure. In the experiments, we obtained, for several classification measures, ML pipelines whose predictive performance was higher than those found by the other tools.

Based on the obtained results, our main contributions can be summarized as follows: (1) The Sleep Spindle data characterization shows that only time-based or spectral features are not suitable for achieving high-performance pipelines; (2) A performance comparison over the F1 score between literature baselines and the GP-based AutoML, showing that the generated pipelines composed of simple ML algorithms can outperform complex *ad hoc* models and; (3) Analysis of the convergence of AutoML algorithms in conjunction with GP features to avoid overfitting, understand pipeline complexity, and reduce data dimensionality.

2. GENETIC PROGRAMMING

GP is an evolutionary computation technique inspired by biological evolution for automatically creating computer programs that solve a given problem. GP iteratively transforms a set (called population) of computer programs through the heuristic search called evolution [Koza 1994].

At each iteration (called generation), GP stochastically modifies a set of possible solutions (called individuals) for the problem, generating new solutions by modifying and recombining the previous set [Poli et al. 2008]. If the algorithm is well parameterized, the most recent individuals will solve the problem better than the previous ones. Moreover, since the technique is stochastic, it is possible to avoid local optima where deterministic methods typically converge [Eberhard et al. 1999].

Individuals are hierarchical tree-like structures composed of primitive functions and terminal values selected by the user for the particular domain of the problem [Poli et al. 2008]. Naturally, in the context of AutoML, the components of the tree structure are data processing functions and ML algorithms.

3. AUTOMATED MACHINE LEARNING

The AutoML pipeline structure is modeled as a Directed Acyclic Graph (DAG). Each node represents a basic algorithm, and the edges represent the input data flow through the selected algorithms. In this way, the size of a particular pipeline, that is, its number of nodes (algorithms), can be used to measure a pipeline's complexity indirectly.

Several widely used open-source AutoML tools are available, and each implements this concept through different strategies. Auto-sklearn, for example, uses Bayesian optimization to find the best algorithms and their hyper-parameters in a pipeline involving preprocessing and modeling functions. These pipelines have a fixed structure formed by the steps of data cleaning, preprocessing, and modeling. In parallel, the framework H2O AutoML uses a random grid search to find the hyper-parameters of a model without adding preprocessing. In the end, the best models are combined using an ensemble.

Finally, the TPOT uses GP to build high-performance pipelines with variable structures for classification or regression problems, providing less overfitting than other tools [Fabris and Freitas 2019]. In this context, the flexible tree structure properties, ease of customization of the nodes, and the very process of evolution of pipelines make GP quite attractive.

The three mentioned frameworks have different strategies for constructing their pipelines when optimizing their solutions. While H2O AutoML generates pipelines with only one predictive layer, Auto-sklearn builds more complex DAGs, despite its fixed structure [Zöller and Huber 2021]. In contrast, TPOT works with DAGs of arbitrary format.

Limiting the universe of pipeline structures reduces the problem's search space. In the case of H2O, which uses a computationally expensive optimization strategy, the generation of simpler DAGs guarantees the evaluation of more solutions [LeDell and Poirier 2020]. Despite the more complex search space, TPOT does not seek an exhaustive evaluation of all combinations of hyper-parameters, as it uses GP to optimize the algorithms.

4. RELATED WORK

Sleep Spindles, visible waveforms in the EEG signal during the polysomnography examination, are very important in sleep staging and identifying pathologies [Devuyst et al. 2011]. Despite a wide variety of recommended features for describing EEG signals [Motamedi-Fakhr et al. 2014], characterization studies of Sleep Spindles data mainly focus on instantaneous statistics of the signal over the spectral and time domains [Ahmed et al. 2009]. In a more general perspective, Lubba et al. [Lubba et al. 2019] present a characterization of multiple time series problems - including some EEG datasets - to find a canonical set of features.

In this kind of application, it is necessary to separate the signal into segments, and for each generated sample, a set of features is calculated. These are varied, but it is possible to define four major groups of features: statistical, spectral, time-frequency, and non-linear [Motamedi-Fakhr et al. 2014]. Despite this, finding a descriptive set to separate the samples into classes cannot be easy. In this context, AutoML can help obtain more generalist pipelines that automatically discover the most relevant features. Our experiments show that we can find less complex and more performant predictive pipelines using more robust characterization measures than literature baselines using fewer data dimensions.

In the literature, several approaches are suggested for the Sleep Spindles data. Tsanas et al. [Tsanas and Clifford 2015] and Zhuang et al. [Zhuang et al. 2016] proposed Continuous Wavelet Transform (CWT) based approaches and the estimation of the probability of spindles occurrences. Lachner-Piza et al. [Lachner-Piza et al. 2018] proposed a Support Vector Classifier (SVC) approach with an *ad hoc* supervised feature selection method based on correlations for determining the importance of each one. Finally, in previous work [Miranda et al. 2019], we presented an automatic feature selection and construction GP-based algorithm to improve simple ML classifiers' performance.

Our previous method uses GP to evolve feature trees to improve the performance of the desired classifier. From the original data, it builds a new dataset composed of combinations of the original features. Our GP combines them through mathematical operators, allowing the addition of non-linearity through truncated mathematical operators, for example. From an AutoML perspective, this method automates attribute selection, creation, and preprocessing. We used the EEG data of Sleep Spindles and K-complexes as an object of study, showing that GP can empower simple classifiers by providing more sophisticated attribute engineering while reducing dimensionality and improving classification results [Miranda et al. 2019].

5. METHODOLOGY

In order to study the application of GP-based AutoML techniques on the EEG classification problem, we performed a series of experiments comparing the performance of the algorithm TPOT with our previous work GP approach for feature engineering [Miranda et al. 2019]. From now on, to facilitate our comparisons, we will name our previous work, Algorithm I. Although both are AutoML techniques and use GP as an optimization strategy, each automates different parts of pipeline creation. While TPOT proposes multi-step automation, our previous work focuses on constructing and selecting attributes.

To carry out our study, we used the Sleep Spindle database of the DREAMS project [Devuyst et al. 2011]. It consists of a collection of signals with expert notes on sleep phenomena or disorders. The dataset has 30-minute 3-channel EEG samples from eight different patients, independently annotated by two sleep specialists.

The EEG signals were segmented into 2-second samples given the maximum duration of 1.67s of the events identified in the data. Using the sliding window method with 75% overlap, samples were extracted from the original signals. For each one, it was decomposed into five levels of Discrete Wavelet Transform (DWT) db5. Finally, four new databases were generated by calculating different sample feature groups.

In Table I, the four groups used and their features are described. The first three (statistical, spectral, and complexity) are already used in applications of EEG sleep signals [Motamedi-Fakhr et al. 2014]. Lastly, the group catch22 comprises 22 features considered canonical for classification problems and time series clustering, obtained through applying multiple feature engineering techniques on 93 different classification problems [Lubba et al. 2019]. The study of this group on EEG data allows comparing the performance of AutoML techniques when evaluating features more or less specific to the application domain.

Table I. Feature groups extracted from EEG data

Group	Features	Signal Domain	# Features
Statistic	{5, 25, 75, 95} percentile, mean, median, variance, entropy, RMS, STD, kurtosis, {zero, mean} crossings	Time	195
Spectral	{5, 25, 75, 95} percentile, mean, median, variance, entropy, RMS, STD, kurtosis, {zero, mean} crossings	Frequency	195
Complexity	{Shannon, FFT, SVD, Fisher} entropy, {Higushi, Pretosian} fractal dimension	Time/Frequency	90
Catch22	Constructed non-trivial time-frequency features	Time/Frequency	330

The spindle identification problem is unbalanced. Less than 2.5% of the total signal from the data was identified as a spindle. To mitigate this problem and maintain the unbalanced nature of the application domain, we reduced the majority class randomly until reaching the proportion of 70% of the samples in all four constructed datasets only for the training and validation sets.

For each group of features, a cross-validation per patient was applied. That is, patients selected for training did not have their samples tested. Thus, each group was performed eight times, with seven patients being used for training and validation and the last for testing.

As techniques based on GP were used in the experiment, the hyper-parameters were kept the same whenever possible: 100 for population size, 100 generations, F1-score for fitness function, 70% for crossover rate, 30% for mutation rate, and a time limit of 100 minutes. The divergent hyper-parameters are shown in Table II. We maintained the same crossover and mutation rates as our previous study for the GP techniques. Algorithm I and TPOT algorithms operate with high mutation rates, given the high amplitude of the search space and the small changes caused by the kind of mutation operator in these approaches.

Table II. AutoML techniques hyper-parameters

	Previous Work	TPOT
Operator set	$\times$, $+$, $-$, $\div$, $\log(\lvert x\rvert)$, $\sqrt{\lvert x\rvert}$	Binarizer, Normalizer, VarianceThreshold, PCA, {MaxAbs, Robust, Standard}Scaler, Select{Fwe, Percentile}, RBFSampler, ZeroCount, FeatureAgglomeration
Classifier	Gaussian NB, DT RF, SV, KNN	Gaussian NB, Bernoulli NB, Multinomial NB, DT, KNN, Logistic regression (LR)
Hiper-parameters	Default (*scikit-learn*)	GP optimization
Terminal set	-0.5, $+0.5$, -1, $+1$	-

In order to observe the advantages of applying AutoML to the problem, we built a comparison basis through training and evaluation of five simple ML classifiers: DT, KNN, Gaussian NB, RF, and SVC. These algorithms were evaluated on the four groups of features without any additional preprocessing. We collected convergence metrics on training, validation, and test data about the pipeline's classification performance (confusion matrix) and complexity (pipeline size and the number of features) during the training. After that, we trained and evaluated Algorithm I and TPOT over the raw groups of features.

Using the performance and convergence metrics of the models, we compared the quality of the pipelines with multiple metrics and models from the literature trained on the same data. In addition, we presented the characterization of the Sleep Spindles on the feature groups, observing the performance and complexity of the pipelines obtained on each of them. Finally, we raised discussions about the dimensionality reduction provided by each algorithm, observing the feature selection and construction operations.

6. EXPERIMENTAL RESULTS

In the literature, we can easily find references that use the DREAMS project databases to study the classification of EEG signals. We selected some references (Table III) with multiple rating metrics (Sensitivity, Recall, Accuracy, and F1 score) to compare with the obtained results. Even though the results are individually competitive with the literature on Recall and Specificity, Algorithm I and TPOT presented a higher average F1 score than the other analyzed references. This shows a better trade-off in rejecting both false positives and false negatives.

To observe the performance of AutoML in contrast to the other classifiers, we compared the F1 score of Algorithm I and TPOT on raw data against ML classifiers on raw data. In Figure 1, we present the performance of each individual classifier or AutoML technique.

For a more direct comparison, Algorithm I (which needs a target classifier selected by the user) was trained on the same classifiers used on raw data. Still, in Figure 1, we can observe that when applying AutoML techniques, there is an increase in the F1 score for all previously tested algorithms, including the case of SVC, which did not converge in the previous step. The TPOT algorithm, which builds

Table III. Comparison of the results obtained with models from the literature

Reference	Recall	Specificity	Precision	$F_1\,score$
[Tsanas and Clifford 2015]	0.76	0.92	0.33	0.46
[Zhuang et al. 2016]	0.51	0.99	0.70	0.59
[Lachner-Piza et al. 2018]	0.65	0.98	0.38	0.48
[Miranda et al. 2019]	0.75	0.98	0.35	0.48
Experiment (Algorithm I)	0.69 ± 0.09	0.92 ± 0.06	0.68 ± 0.19	0.65 ± 0.11
Experiment (TPOT)	0.67 ± 0.07	0.95 ± 0.03	0.73 ± 0.18	0.68 ± 0.09

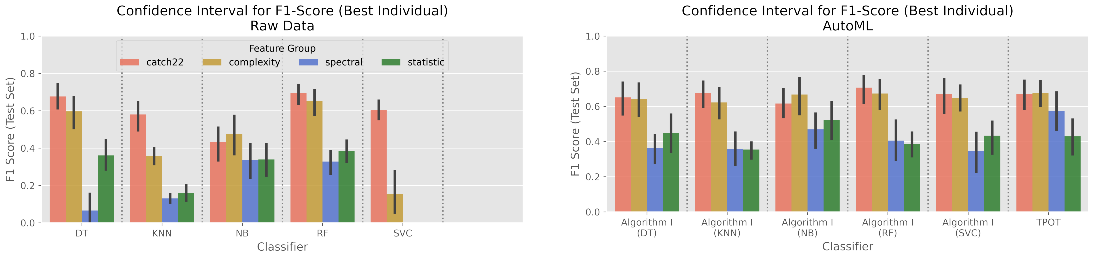

Fig. 1. Evaluation of traditional ML models on raw data (left) and F1 score obtained from AutoML pipelines on feature groups through cross-validation (right). Each classifier or AutoML technique was evaluated on the four groups of features (represented in different colors) with the metric F1 score, represented by the vertical bars and their respective confidence intervals (calculated over the cross-validation results).

more general pipelines with less focus on feature engineering, obtained competitive results with the others. In addition, it stands out in the use of the features of the Spectral group.

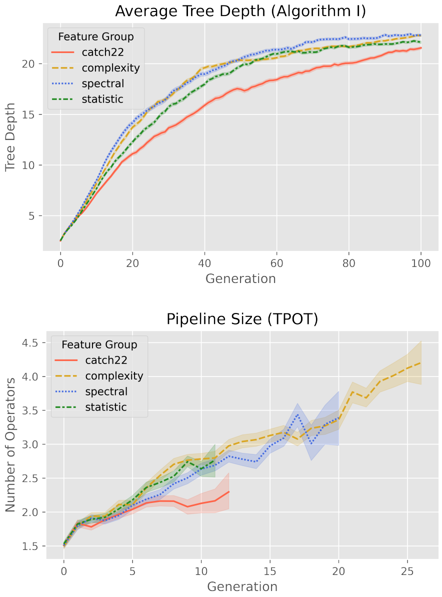

Fig. 2. Increase in the complexity of individuals (pipelines) over generations.

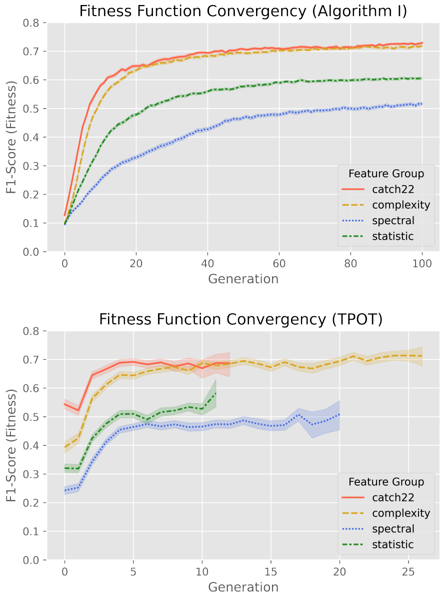

Fig. 3. Convergence of the fitness function (population) on the validation data

When analyzing the evolution of the fitness function in Figure 3, we can observe a sharp growth in the initial generations followed by a slight increase until the end of the execution. Due to the established limit of 100 minutes, TPOT interrupted its execution before 100 generations. Both techniques converge to low values of F1 score in the Spectral and Statistical groups, showing that the algorithms have difficulty adjusting pipelines capable of differentiating the classes of the problem. On the other hand, the Complexity and Catch22 groups proved to be more suitable for characterizing the problem.

To observe the increase in pipeline complexity, we analyzed the average number of operators in the population at each generation for TPOT and its equivalent for Algorithm I, represented by the average depth of individuals. Figure 2 shows the evolution of the average complexity of the pipelines for each feature group over the generations.

With the increase in fitness, we can also observe an increase in the complexity of the pipelines in Figure 2. To obtain solutions with higher performance, GP increments the pipelines with more features, in the case of Algorithm I, and with more operators in the case of TPOT. However, the increase in complexity does not always bring relevant improvements in performance. By observing the best trade-off between pipeline complexity and individual fitness, we can define a criterion for early stopping the algorithm.

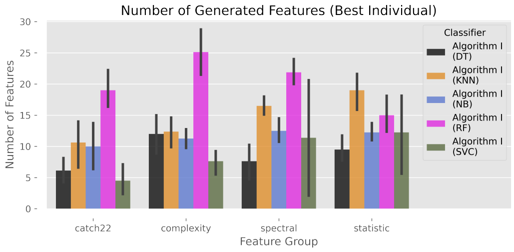

Fig. 4. Number of features per group obtained with Algorithm I.

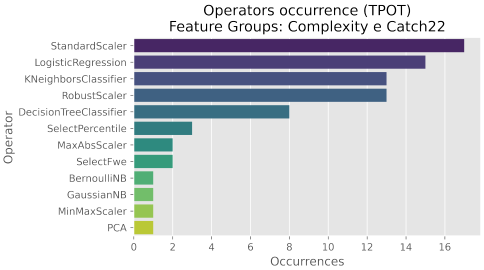

Fig. 5. Operators found in the pipelines generated by TPOT.

Although Algorithm I and TPOT are AutoML techniques, each automates different ML tasks. When analyzing the content of the constructed pipelines, Algorithm I provides a high dimensionality reduction for all feature groups regardless of the classifier used, as shown in Figure 4. In this Figure, the grouped bar chart shows the average number of attributes generated for each group of attributes through Algorithm I. As this technique uses a target classifier, the number of constructed attributes may vary according to the choice. Despite this variation, Algorithm I can bring up to a 97% reduction in dimensionality compared to the number of attributes of the raw data presented in Table I.

For TPOT, it is not possible to measure dimensionality reduction so explicitly. However, as an indirect measure, in Figure 5, we present the number of occurrences per algorithm in the pipelines. So, it is possible to observe the TPOT's most frequent choices for the classification problem. When analyzing the complexity groups and catch22, TPOT selects a few feature reduction operators, such as PCA, *SelectPercentile*, and *SelectFwe* - predominantly, so its solutions work with all dimensions.

7. CONCLUSIONS

Time series analysis applications, particularly EEG signals, are challenging ML applications. For the analysis of EEG signals, the signals need to be segmented. Signal segmentation requires a pipeline of several stages when various transformations, cleaning functions, and, mainly, feature extraction techniques. Optimizing the pipeline in this variety of tasks generates a complex search space to be explored, especially for data scientists who do not have in-depth knowledge of the application domain.

In this comparative study, we investigated through systematic experiments the characterization of Sleep Spindle data, the performance comparison on the F1 score between literature baselines and GP-based AutoML, and the analysis of the convergence of the algorithms to avoid overfitting, understand pipeline complexity, and reduce data dimensionality. The results showed that the use of AutoML on the sleep EEG signals classification problem can generate more accurate solutions than complex models in the literature and also quantifies the importance of each group of features used. Furthermore, the analyzed techniques Algorithm I and TPOT can build more accurate classifiers than models from the literature through simple operators and classifiers. In addition, we collected evidence that using complexity measures and constructed features can obtain a better characterization of sleep spindles, improving the performance of the classifiers used in the raw data.

There is still room to improve strategies from GP to AutoML. TPOT could not evolve its pipelines by the desired number of generations because it reached the time limit. In addition, the algorithm was interrupted in some executions due to inadequate management of computational resources. Despite superior results to models in the literature, their solutions can perform only simple transformations on the data. In parallel, Algorithm I performs a complex feature engineering and dimensionality reduction. However, its pipelines use only one classifier and cannot select preprocessing functions.

REFERENCES

Ahmed, B., Redissi, A., and Tafreshi, R. A characterization of sleep spindles in eeg. In *World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany.* Springer Berlin Heidelberg, 2009.

Azevedo, A. I. R. L. and Santos, M. F. Kdd, semma and crisp-dm: a parallel overview. *IADS-DM*, 2008.

Bontempi, G., Taieb, S. B., and Le Borgne, Y.-A. Machine learning strategies for time series forecasting. In *European business intelligence summer school.* Springer, pp. 62–77, 2012.

Devuyst, S., Dutoit, T., Stenuit, P., and Kerkhofs, M. Automatic sleep spindles detection—overview and development of a standard proposal assessment method. In *2011 Annual international conference of the IEEE engineering in medicine and biology society.* IEEE, pp. 1713–1716, 2011.

Eberhard, P., Schiehlen, W., and Bestle, D. Some advantages of stochastic methods in multicriteria optimization of multibody systems. *Archive of Applied Mechanics* 69 (8): 543–554, 1999.

Fabris, F. and Freitas, A. A. Analysing the overfit of the auto-sklearn automated machine learning tool. In *International Conference on Machine Learning, Optimization, and Data Science.* Springer, pp. 508–520, 2019.

Hutter, F., Kotthoff, L., and Vanschoren, J. *Automated Machine Learning.* Springer, 2019.

Kevric, J. and Subasi, A. Comparison of signal decomposition methods in classification of eeg signals for motor-imagery bci system. *Biomedical Signal Processing and Control* vol. 31, pp. 398–406, 2017.

Koza, J. R. Genetic programming as a means for programming computers by natural selection. *Statistics and computing* 4 (2): 87–112, 1994.

Lachner-Piza, D., Epitashvili, N., Schulze-Bonhage, A., Stieglitz, T., Jacobs, J., and Dümpelmann, M. A single channel sleep-spindle detector based on multivariate classification of eeg epochs: Mussdet. *Journal of neuroscience methods* vol. 297, pp. 31–43, 2018.

LeDell, E. and Poirier, S. H2O AutoML: Scalable automatic machine learning. *7th ICML Workshop on Automated Machine Learning (AutoML)*, July, 2020.

Lubba, C. H., Sethi, S. S., Knaute, P., Schultz, S. R., Fulcher, B. D., and Jones, N. S. catch22: Canonical time-series characteristics. *Data Mining and Knowledge Discovery* 33 (6): 1821–1852, 2019.

Miranda, Í. M., Aranha, C., and Ladeira, M. Classification of eeg signals using genetic programming for feature construction. In *Proceedings of the Genetic and Evolutionary Computation Conference.* pp. 1275–1283, 2019.

Motamedi-Fakhr, S., Moshrefi-Torbati, M., Hill, M., Hill, C. M., and White, P. R. Signal processing techniques applied to human sleep eeg signals—a review. *Biomedical Signal Processing and Control*, 2014.

Olson, R. S. and Moore, J. H. Tpot: A tree-based pipeline optimization tool for automating machine learning. In *Automated Machine Learning.* Springer, pp. 151–160, 2019.

Poli, R., Langdon, W. B., and McPhee, N. F. *A field guide to genetic programming*, 2008.

Tsanas, A. and Clifford, G. D. Stage-independent, single lead eeg sleep spindle detection using the continuous wavelet transform and local weighted smoothing. *Frontiers in human neuroscience* vol. 9, pp. 181, 2015.

Zhuang, X., Li, Y., and Peng, N. Enhanced automatic sleep spindle detection: a sliding window-based wavelet analysis and comparison using a proposal assessment method. In *Applied Informatics.* Vol. 3. SpringerOpen, 2016.

Zöller, M.-A. and Huber, M. F. Benchmark and survey of automated machine learning frameworks. *Journal of Artificial Intelligence Research* vol. 70, pp. 409–472, 2021.