Automated Grammar-based Feature Selection in Symbolic Regression

Muhammad Sarmad Ali
University of Limerick
Limerick, Ireland
sarmad.ali@ul.ie

Meghana Kshirsagar
University of Limerick
Limerick, Ireland
meghana.kshirsagar@ul.ie

Enrique Naredo
University of Limerick
Limerick, Ireland
enrique.naredo@ul.ie

Conor Ryan
University of Limerick
Limerick, Ireland
conor.ryan@ul.ie

ABSTRACT

With the growing popularity of machine learning (ML), regression problems in many domains are becoming increasingly high-dimensional. Identifying relevant features from a high-dimensional dataset still remains a significant challenge for building highly accurate machine learning models.

Evolutionary feature selection has been used for high-dimensional symbolic regression using Genetic Programming (GP). While grammar-based GP, especially Grammatical Evolution (GE), has been extensively used for symbolic regression, no systematic grammar-based feature selection approach exists. This work presents a grammar-based feature selection method, Production Ranking based Feature Selection (PRFS), and reports on the results of its application in symbolic regression.

The main contribution of our work is to demonstrate that the proposed method can not only consistently select the most relevant features, but also significantly improves the generalization performance of GE when compared with several state-of-the-art ML-based feature selection methods. Experimental results on benchmark symbolic regression problems show that the generalization performance of GE using PRFS was significantly better than that of a state-of-the-art Random Forest based feature selection in three out of four problems, while in fourth problem the performance was the same.

KEYWORDS

grammatical evolution, symbolic regression, feature selection, grammar pruning, production ranking

ACM Reference Format:
Muhammad Sarmad Ali, Meghana Kshirsagar, Enrique Naredo, and Conor Ryan. 2022. Automated Grammar-based Feature Selection in Symbolic Regression. In *Genetic and Evolutionary Computation Conference (GECCO '22), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3512290.3528852

GECCO '22, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9237-2/22/07.
https://doi.org/10.1145/3512290.3528852

1 INTRODUCTION

The success of machine learning applications and the growing trend of learning from data has given rise to the collection of more and more data in almost every domain. Many industrial and real-world applications demand accurate predictions from numerical data. Regression, or symbolic regression, aims to learn a model from the numerical data to predict a target. However, real-world regression datasets are not clean and are usually contaminated with various types of noise [14], for instance, measurement noise from the sensors, rounding-off noise in manual readings, etc. In addition, there can be redundancy in the data, where multiple features measure the same attribute in the same environment. Moreover, it has been established that not all features from a data set contribute equally toward predicting the output. Generally, a subset of features describes much of the output; these are more predictive and thus more important or relevant for learning. Identification of such 'relevant' features is a well-know problem [6]. As the number of features increase in a dataset, it becomes more challenging to identify relevant features due to the associated problems of scalability, performance and the 'curse of dimensionality' [11].

In the past few decades, there has been a lot of research attention on feature selection in machine learning. Evolutionary feature selection has also gained research focus and several evolutionary algorithms have been utilized including Genetic Programming (GP) [13]. GP and its variants inherently perform feature selection while evolving candidate solutions. However, this selection is not explicitly guided, rather it is an emergent (and stochastic) property of the algorithm, because a population implicitly acquires knowledge about relevant features as individuals using them are more likely to achieve higher fitness. Most of the GP-based evolutionary feature selection approaches exploit this knowledge.

Grammatical Evolution (GE) [30] is an approach to genetic programming that uses a context free grammar (CFG) to specify the space of possible solutions. When performing symbolic regression with GE, function primitives and features are represented as grammar *terminals*, which are combined using *production rules*. Grammars are highly expressive and enable users to group various functions, terminals and features together in logical ways. A grammar-based approach to feature selection would result in a refined grammar.

In this paper, we present a grammar-based feature selection method and report on the results of its application to symbolic

regression. To our knowledge, it is the first grammar-based feature selection method. When developing the method, our objective was not only to identify relevant features, but to investigate how our approach impacts the generalization performance in GE, since generalization improvement is an open research agenda [26]. Therefore, we posed following research questions:

RQ1: Can the grammar-based feature selection method select relevant features for symbolic regression?

RQ2: How does this feature selection affect generalization performance in GE as compared to feature selection through common ML methods?

To answer these two questions, we designed two sets of experiments, the details of which are presented in section 4. In the next section, we briefly discuss existing research contributions which are closely related to our work. We then present the outlines of our proposed method in Section 3, and the results are discussed in section 5.

2 RELATED WORK

There is a huge body of research contributions both in ML and GP addressing the problems of feature selection and regression. Since our work focuses on evolutionary feature selection and symbolic regression using GE, we briefly discuss closely related existing efforts in these research directions below.

2.1 GE for Symbolic Regression

Symbolic regression searches the space of mathematical expressions to find a model that best fits the given dataset. GE has been extensively used to evolve expressions by randomly combining building blocks (mathematical operators, functions, constants, and variables) which are represented in the grammar. A good overview can be found in [31]. Although several works considered standard GE, there are notable efforts for using enhanced GE methods for symbolic regression, for example Structured GE [16], Geometric Semantic GE [21], π-GE [25], Hierarchical GE [18]. In addition to using simple CFGs, there have been efforts to exploit varying grammar formalisms, for example GE with stochastic CFG [19], attribute grammar [28], tree-adjoining grammar [22], and Christiansen grammar [27]. While our work focus on feature selection in symbolic regression, there are a few efforts on function set selection as well in grammar-based GP. Recent work by Ali et al. [2, 3] examined the effect of various grammar structures and grammar pruning for function set selection in symbolic regression. Nicolao and Agapitos [24] studied effect of function set selection on generalization performance of GP and GE on symbolic regression problems.

2.2 Evolutionary Feature Selection

Feature selection (FS) is performed to remove irrelevant and redundant features and enhance the learning performance of the ML algorithm. However, for high-dimensional problems, many existing FS methods either become cost ineffective or suffer from stagnation in local optima [38]. Evolutionary computation approaches to FS gained attention due to their global search ability. Xue et al. [38] provided a nice review of evolutionary approaches for FS.

There are a number of studies which perform feature selection using GP. Though a vast majority consider classification problems, a few investigated applications to symbolic regression. Smit et al. [34] proposed a Fitness-Inheritance approach for variable/feature selection using Pareto-GP. With the aim to build robust models from industrial datasets, they perform non-linear sensitivity analysis and accumulate feature ranks for all the solutions on Pareto front. Chen et al. [7] presented an evolutionary FS approach, GPWFS, for high-dimensional symbolic regression. Using a 2-stage architecture, they achieved better performance both on training and test sets. Chen et al. [8] presented another approach, GPPI, which was based on the idea of permuting feature vector to assess its impact on regression error. With a detailed comparison with GPWFS, its variants, and two ML methods, they concluded that their approach resulted in better generalization performance. Helali et al. [1] used GP as a context-based selection mechanism. The selection of features was determined by the change in the performance of the evolved GP models when the feature were injected with noise. The approach was applied to select imputation predictors in symbolic regression for problems with missing values.

Gavrilis et al.[10] exploited GE's inherent feature selection capability, but put more emphasis on feature construction. Over a mix of classification as well as regression problems, feature transformation functions were evolved to create new features/datasets. Silva and Leong [33] also focused mainly on feature generation with GE. Selection was performed to eliminate generated features that were irrelevant and to improve the model accuracy. A recent work by Monteiro et al. [20] proposes a feature engineering solution using a GE variant. Structured Grammatical Evolution [16], composing original and generated features. Their strategy is a combination of feature selection and generation. In all of the above studies, the selection is solely based on GE's inherent feature selection ability which is not strong enough since it does not consider feature's relevance [8].

3 GRAMMAR-BASED FEATURE SELECTION

Features are typically represented as terminal productions[1] in the grammar. In a grammar-based approach, selecting relevant features would mean identifying relevant terminals and removing irrelevant ones from the grammar. Our approach is fully automated and is based on a generic 2-stage architecture, the *X-GE* system ('X' refers to a feature selection method). As shown in Figure 1, Stage 1 is the feature selection (FS) stage where any FS method can be plugged in to identify relevant features. In Figure 1, Stage 1 comprises of our proposed grammar based feature selection approach, *Production Ranking-based Feature Selection* (PRFS). We evolve candidate solutions for a small number of g generations. At the end of each generation, a subset of individuals in the population are structurally analyzed and assigned *ranking* scores based on the frequency count of productions. The production ranking scores are accumulated across generations and high ranked productions are selected to be retained in the grammar. Stage 2 is *Grammatical Evolution with Feature Selection* (GEFS), where terminals corresponding to selected

[1] In the grammar, there are also non-terminal productions, as well as other terminal symbols which do not represent features/variables. However, since we are focusing discussion on features, use of the word 'terminals' would refer to terminal productions in the grammar that represent features.

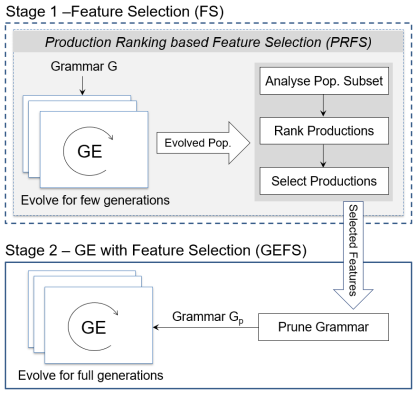


Figure 1: The X-GE system architecture. An X-GE instance, PR-GE system, uses PRFS at Stage 1.

features, either through PRFS or using some other feature selection method (see section 4.2), are retained while the rest are *pruned* from the grammar. Overall, our approach is based on the key steps of structural analysis, production ranking and grammar pruning, which are explained next.

3.1 Structural Analysis

In GE, every individual in the population is composed of terminal symbols which appear in an order defined by the derivation tree constructed during genotype to phenotype mapping. By traversing the derivation tree, it is possible to obtain a list of grammar productions used in the mapping process to generate an individual. Such a list is referred to as the *production-list*. Once identified, the frequency of usage of each production in the production-list can be easily determined.

An important consideration is to decide how much of the population to analyze and select for ranking. We experimented with three possible choices: 1) the entire population, 2) only unique individuals, 3) top $K\%$ of the population. Based on our empirical evaluations, the third option turned out to be the best choice when ranking productions containing only terminal symbols. We also examined a variety of values for K. A lower value of K would analyze only few good individuals and there is a risk of losing important productions that had not yet enjoyed high fitness, while too high of a value would increase the computational overhead. Through an extensive set of experiments for hyperparameter search (see Section 4.3), we found $g = 5$ and $K = 20\%$ to be the best choice in our case. In [7, 8], top 5% to 15% was considered, though they use a much higher value of g (50 or more).

3.2 Production Ranking

Productions can be ranked based on how frequently they are used in the construction of individuals in the population. As evolution proceeds, fitter individuals survive, and the productions which more frequently shape the structures are the ones that are considered to be worthy being part of the grammar. Such productions are assigned a high rank. On the contrary, productions which harm individual's fitness such that they become extinct, generally do not enjoy high usage frequency (although rarely zero, due to hitch-hiking effects) in the population. Below, we provide a formal account to our ranking scheme.

Let P be the set of productions in the grammar G. $P_i \subset P$ is the set of productions in the production-list of the ith individual. If $n = |P|$, number of productions in the P, and $k = |P_i|$, then k < n for practically all individuals. The ranking score assigned to the jth production in the production-list of ith individual is given by:

$$(nfr)_i^j = \left(\frac{\phi_i^j}{l_i} \right) \tag{1}$$

$$(fpr)_i^j = (nfr)_i^j \times \rho_i \tag{2}$$

where ϕ_i^j is the frequency of jth production, l_i is the effective codon length, and ρ_i is the fitness of ith individual. Equation 1 defines the *normalized frequency rank* (*nfr*) of a production, while Equation 2 computes the *fitness-proportionate rank* (*fpr*). As a consequence of the above two definitions, the following two properties hold for an ith individual:

$$\sum_{j=1}^{k} (nfr)_i^j = 1 \quad \text{and} \quad \sum_{j=1}^{k} (fpr)_i^j = \rho_i$$

Let u be the number of individuals in the population subset. Once ranking scores of each production in the production-list have been computed for all u individuals, we accumulate the scores to compute *generation worth* (*gw*) for a single generation. We then accumulate *gw* across all g generations to compute the overall *run worth* (*rw*), as given in the following equations:

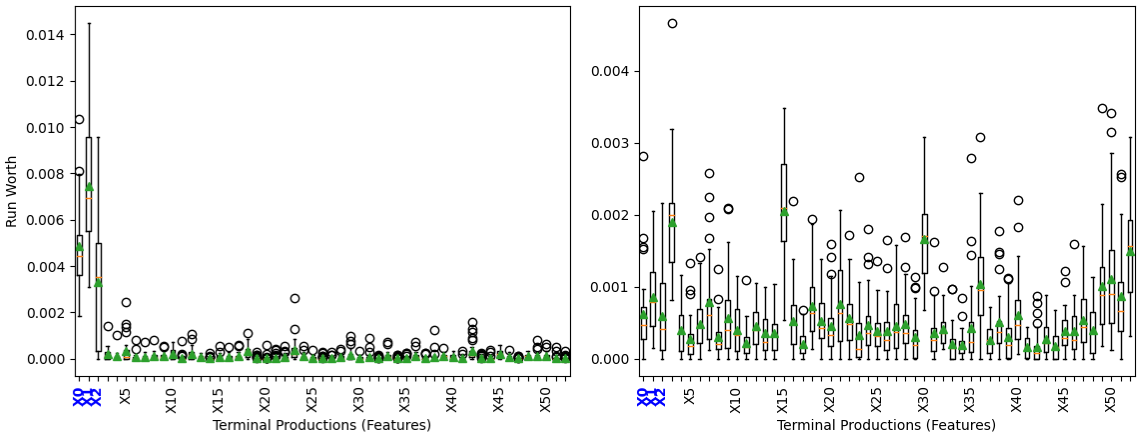
$$(gw)_m^j = \sum_{i=1}^{u} (fpr)_i^j \tag{3}$$

$$(rw)^j = \sum_{m=1}^{g} (gw)_m^j \tag{4}$$

Finally, the run worth of each production is averaged over all the runs (30 in our case). To minimize the computational cost of production ranking, we track the production-list during the mapping process, although it does incur a small memory overhead. However, since the ranking scores are computed at the end of an evolutionary run, and the operations defined by the above equations are trivial, those can be efficiently performed with minimal overhead.

3.3 Production Ranking with Fitness Scaling

While computing *fpr* in Equation 2, our ranking system can either use raw fitness or linearly scaled fitness. Since linear scaling relieves GP of the coefficient search, GP can focus on finding "*expression whose shape is most similar to that of the target function*" [12]. We hypothesized that finding the 'right' shape would let GP form expressions with the 'right' ingredients. To test our hypothesis, we compared production ranks using both raw and linearly-scaled fitness. The two example box plots of the production ranks, called *PR-plots* of the terminal productions in the grammar against Exp 1.1 (see Table 4) are show in Figure 2. The PR-plot (a) and (b) are generated using raw and linearly-scaled fitness respectively for the same experiment. It is evident that using linearly-scaled fitness when computing production ranks can award significantly high ranks to some productions. The high rank productions correspond to relevant features while productions with significantly low ranks



(a) With LS (b) Without LS

Figure 2: PR-plots for Exp 1.1 (see Table 4) showing effect of Linear Scaling (LS) on production ranking. With LS the more salient (relevant) features are correctly identified, whereas without LS it is virtually impossible to distinguish the features.

are irrelevant ones. Using raw fitness could not distinguish among features. It is worthwhile to mention that with linear-scaling PRFS was always able to correctly distinguish the most salient features in just 5 generations, whereas with raw fitness it took 20 generations or more. Hence, the small computational overhead of linear-scaling brings in huge benefits of accuracy and consistency (more in section 5.3).

3.4 Production Selection & Grammar Pruning

The higher the rank of a production, the more is the importance of the associated feature. At the end of PRFS (stage 1), we select the high rank terminal productions. The selection criteria can be defined by domain experts or though some statistical measure. Through empirical investigation (see 4.2.1), we chose to select productions whose production ranks are outliers in the list of ranks. With the selected terminal productions we proceed to Stage 2 (GEFS) and present those productions to the *Grammar Pruning* module, which is responsible to update the grammar by pruning irrelevant productions. As shown in Figure 1, the updated grammar G_p is input to GE in stage 2 to proceed with regular evolutionary trials.

4 EXPERIMENTAL SETUP

This section details our experimental design. We describe the datasets used, the values of the evolutionary parameters, the grammar and the ML-based FS methods used for comparative analysis.

4.1 Dataset

We utilized two types of dataset based on the kind of experiments we conducted to answer each research question. In *RQ1*, we are interested to see if our approach can identify *truly relevant* features (TRFs). Following the approach taken in [8] to demonstrate selection of truly relevant features, we used two synthetic functions for which the actual input variables are known and added a large number of irrelevant *artificial* features (AFs) to the original data. We exercised PRFS on the newly generated dataset with many irrelevant features to rank and identify top k features, where k is the number of truly relevant features. Note that PRFS identifies as many features as it deems salient and that k is not a parameter.

Table 1 lists the actual functions, input ranges for the actual features, and the number of data samples. The first function F_1 is Newton's law of universal gravitation having three input variables (m_1, m_2, r) and the gravitational constant G. This function is also used in [8], with the same input ranges. However, to pose a harder problem for our approach, we chose Vladislavleva-4 benchmark function [36] as our second function, F_2, which contains 5 input variables. We further complicate the problem of identifying truly relevant features with a number of changes which we detail below in Section 5.1.

Our second investigation, in *RQ2*, pertains to generalization performance through feature selection. For that, we chose four real-world regression datasets which contain from relatively low (25) to high (~200) numbers of input features. Table 2 lists the problems considered in this work; these come from a variety of application domains including chemical processing, law-enforcement, and drug design. The chosen datasets have been used in earlier research work on evolutionary feature selection using GP [7, 34]. Each dataset is assigned a short name (in distinct font) that will be used to refer in subsequent sections.

A few details are worth mentioning here. The `tower` dataset was collected from PonyGE2 github repository[2], while `dowchem` dataset was acquired from GPBenchmarks.org[3] website. Both datasets were used without any preprocessing. The `ccrime` dataset was acquired from UCI Machine Learning repository. We removed non-predictive and features with missing values, which left the dataset with 100 input features. The `bioava` dataset was obtained from the authors of [4]. There are 47 features which contain a value of zero across all samples. Since those features provide no useful information for learning [9], we removed those in pre-processing, leaving the dataset with 194 features.

[2]https://github.com/PonyGE/PonyGE2/tree/master/datasets
[3]http://gpbenchmarks.org/?page_id=30

Table 1: Synthetic Functions. U[a,b,c] means c number of uniform random samples drawn from a to b inclusive.

Function	Input Ranges
$F_1 = G\frac{m_1 m_2}{r^2}$	$m_1, m_2 : U[0, 1, 100] \quad r : U[1, 2, 100]$
$F_2 = \frac{10}{5 + (x-3)^2 + (y-3)^2 + (z-3)^2 + (v-3)^2 + (w-3)^2}$	$x, y, z, v, w : U[0.05, 6.05, 1024]$

Table 2: Datasets used in experimentation

Dataset	Short name	Features	Instances
Tower	`tower`	25	4999
Dow Chemical	`dowchem`	57	1066
Communities and Crime	`ccrime`	128	1994
Human Oral Bioavailability	`bioava`	241	359

4.2 Feature Selection Methods

Our problem domain is (multivariate) regression where the target (to be predicted) is numeric, so are the input variables/features. We therefore employed several popular ML-based feature selection approaches used in regression as an alternative to PRFS in Stage 1 in our system (see Figure 1). Each of the FS methods were plugged into the X-GE architecture to instantiate the corresponding X-GE method (where X refers to the feature selection approach listed below).

- *Corr-FS*: This method relies on selecting numeric predictors using the correlation statistic (usually Pearson's Correlation). The Corr-FS combined with GEFS is referred to as **Corr-GE**.
- *MI-FS* refers to Mutual Information based feature selection. The corresponding X-GE method is termed **MI-GE**.
- *SF-FS* is the Sequential Forward Feature Selection, a greedy procedure which iteratively keeps adding the feature that best improves the model performance. The **SF-GE** is the corresponding X-GE method.
- *PFI-FS* refers to Permutation Feature Importance based FS. It measures the decrease in a model performance when a single feature value is randomly permuted. This approach is used by Random Forest over out-of-bag (OOB) examples and is one of the state-of-the-art approaches for feature selection. **PFI-GE** method combines PFI-FS with GEFS.

We used *scikit-learn* Python library to perform feature selection tasks using default parameters for various regressor instances. Each of the above mentioned X-GE method was compared with PR-GE for impact on generalization performance for the real-world regression problems shown in Table 2.

4.2.1 How Many Features to Select? There is no standard measure to select features from the list of feature importance. One approach is incremental search, but it becomes increasingly expensive as the number of features increase. For high-dimensional datasets, this is not feasible. The other common approach is to pick top K features, where K is usually less than 50%. A high value of K would compromise the advantages of efficiency, while a low value may not achieve the accuracy targets.

Table 3: Experimental Settings

Method	Hyperparameter	Value
PRFS	Population Size	500
	Pop. subset for Ranking (K)	Top 20%
	Number of Generations (g)	5
	Linear Scaling	Enabled
GE/GEFS	Population Size	250
	Number of Generations	100
	Linear Scaling	Disabled
-common-	Number of Runs	30
	Search Engine	Steady-State GA
	Sampling Method	Random Sampling
	Train/test split	70/30
	Crossover Type	Effective Crossover [5]
	Crossover Probability	0.9
	Mutation Probability	0.01
	Selection Type	Tournament
	Initialization Method	Sensible Initialization [29]

With the intention to use some statistical measure for feature selection in our work, we ran Shapiro-Wilk Normality Test [32] on production ranks over a large number of empirical trials using various datasets. It was found that the ranks are not normally distributed. Furthermore, we also found that the best set of features are the ones whose production ranks are outliers in the vector of production ranks, i.e. greater than 1.5× (75^{th} percentile). Results of those trials are not presented in this work due to space limitations. All X-GE experiments in this work used the same cardinality of features as defined by PRFS based on the number of outlier ranks corresponding to the given dataset.

4.3 Parameters

Table 3 presents the evolutionary parameters used in all experimental runs. We performed random subsampling cross validation with a repeat factor of 3. For repetition we used a different seed to ensure a different training-test data split was generated each time. The repetition ensured that we further minimized the chances of bias and overfitting [37]. Each experiment was run 30 times to ensure that any conclusions drawn from the results were statistically sound [17]. Note that linear scaling was not exercised during GEFS stage.

The search for good parameters for PRFS approach was carried out empirically by performing a grid hyperparameter search. A number of experiments were conducted on a few synthetic functions and real-world datasets by varying the number of generations and population in the set [5, 10, 15, 20] and [250, 500, 750, 1000]

respectively. The results of those experiments are not presented due to space constraints. It was found that evolving PRFS runs for 5 generations with a population size of 500 was a good, although not necessarily the best, choice.

The fitness function measures the performance of the algorithm against a predefined objective. Since our problem domain is symbolic regression, we used the common fitness function, Root Mean Squared Error (RMSE), which is defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{5}$$

where n is the number of data points, y_i is the target value, and $\hat{y}_i$ is the predicted value. RMSE assesses the mean extent of deviation from the desired value, so the goal of evolution is to minimize this error metric across generations. Note that both X-GE system stages utilize RMSE as fitness function.

4.4 Grammar

The grammar embodies all the features as well as function primitives as productions. It has been reported in [2, 23] that using the *extended* function set that includes many trigonometric, exponential and power functions can result in better generalization performance. Therefore, the grammar used in all executions of PRFS and GEFS (except in a small number of experiments, see Section 5.1), is the *balanced* version of the *extended grammar* defined in [2]:

```
 <expr> ::= <expr1> | <var> | <expr2> | <var>
<expr1> ::= sin(<expr>) | cos(<expr>) | tan(<expr>)
        | pow(<expr>,2) | pow(<expr>,3) | sinh(<expr>)
        | cosh(<expr>) | tanh(<expr>) | exp(<expr>)
        | exp(-1*(<expr>)) | log(abs(<expr>))
        | sqrt(abs(<expr>)) | (-1*(<expr>))
        | (1/<expr>)
<expr2> ::= (<expr> - <expr>) | (<expr> * <expr>)
        | (<expr> + <expr>) | (<expr> / <expr>)
  <var> ::= X0 | X1 | X2 | X3 | X4 | ...
```

We defined two other versions of the grammar: basic, and arithmetic. The *basic grammar* excluded productions corresponding to the functions $x^2, x^3, sinh, cosh, tan, e^x$, and e^{-x}, while *arithmetic grammar* only contained arithmetic operators (excluding the rule <expr1> and corresponding productions).

5 RESULTS & DISCUSSION

We now present the outcome of the extensive set of experiments that we conducted to verify that PRFS is not only capable of identifying relevant features (Section 5.1), but also has the potential to outperform state-of-the-art feature selection methods when performing symbolic regression with GE (Section 5.2). In Section 5.3 we detail how we empirically assessed that PRFS can consistently identify almost similar set of relevant features.

5.1 Selection of Truly Relevant Features

To validate if our approach can select truly relevant features (TRFs), we ran multiple experiments on the two synthetic functions listed in Table 1. Each experiment concluded after the ranking stage with the default parameters (see Table 3). We evolved candidate solutions for 5 generations with linear scaling and a population size of 500, with 30 independent trials in each experiment. Random sampling was used with random 70/30 data split between training and test, using trial number as a seed. Production ranks were averaged out over 30 trials and features for whom the production ranks appeared as outliers were selected.

Table 4: Experiments to verify if PRFS can select Truly Relevant Features (TRFs) when mixed with 50 Artificial Features (AFs).

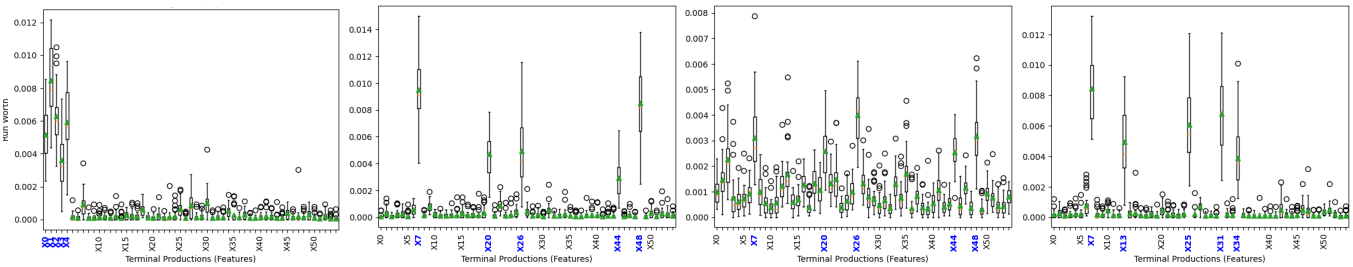
Func.	Exp	Grammar	Index of TRFs	Range (AFs)	Selected?
F1	1.1	Extended	0, 1, 2	[0, 1]	Yes
	1.2	Basic	0, 1, 2	[0, 1]	Yes
	1.3	Arithmetic	0, 1, 2	[0, 1]	Yes
F2	2.1	Extended	0, 1, 2, 3, 4	[0, 1]	Yes
	2.2	Extended	0, 1, 2, 3, 4	[0.05, 6.05]	Yes
	2.3	Extended	rnd (7, 20, 26, 44, 48)	[0.05, 6.05]	Yes
	2.4	Basic	rnd (7, 20, 26, 44, 48)	[0.05, 6.05]	Yes
	2.5	Arithmetic	rnd (7, 20, 26, 44, 48)	[0.05, 6.05]	Yes
	2.6	Extended	rnd (7, 13, 25, 31, 34)	[0.05, 6.05]	Yes

Details of the experiments and the outcomes are presented in Table 4. The first and second columns show the synthetic function and the experiment number respectively. The third column lists which grammar version was used (see Section 4.4). The fourth column indicates the index of the truly relevant features in the generated dataset along with 50 artificial features. The range of values for artificial features is shown in fifth column. The final column shows if the TRFs appeared as top 3 (for $F1$) or top 5 (for $F2$) features among the set of selected features (4-7 for F1 and 5-8 for F2.

5.1.1 Experiments on F1. Three experiments were conducted on $F1$ with different primitive sets modeled in their respective grammars. The corresponding PR-plot for Exp 1.1 is shown in Figure 2(b). It is visually evident that the top 3 selected features included the TRFs ($X0$, $X1$, and $X2$). The ranks of TRFs are significantly higher than the ranks of AFs. The PR-plots against Exp 1.2 and Exp 1.3 were very similar, so omitted due to space constraints.

5.1.2 Experiments on F2. On the Vladislavleva-4 benchmark function, we conducted six experiments, each with different setups. In the first experiment, the input range for the AFs was [0, 1]. However, it could be trivial for GE and ranking algorithm to identify TRFs due to the different range. Therefore, in the second experiment we kept the same input range for AFs as that of TRFs. The PRFS algorithm was able to correctly select the top 5 features in both experiments. The PR-plot for Exp 2.2 is shown in Figure 3(a). The plot for Exp 2.1 was very similar and was therefore omitted.

For the next three experiments (Exp 2.3 - 2.5), we made the problem more challenging by randomly placing the TRFs within 50 AFs. In each experiment, we used different function primitives in the grammar. PRFS was able to correctly identify the TRFs in all three experiments. The PR-plots for Exp 2.3 and Exp 2.5 are shown in Figure 3(b) and 3(c). The plot for Exp 2.4 was very similar to that of Exp 2.3, so it is not presented. The plot against Exp 2.5 is unique in the sense that the productions corresponding to AFs received much higher ranks when compared to other experiments on $F2$. Even though the first 5 high-ranked productions were the



(a) Exp 2.2 (b) Exp 2.3 (c) Exp 2.5 (d) Exp 2.6

Figure 3: PR-plots for experiments on *F2* (see Table 4). The TRFs (shown in blue) carry much higher production ranks as compared to AFs.

Table 5: Comparative results of GE vs various FS methods

Dataset	GE	Corr-GE	MI-GE	FFS-GE	PFI-GE	PR-GE
tower	63.24 (2.87)	62.84 (1.27) =	64.19 (3.17) =	60.45 (3.61) +	58.24 (3.47) +	***56.21 (5.19)*** **+**
dowchem	0.334 (0.017)	**0.310 (0.032) +**	0.341 (0.017) =	0.334 (0.018) =	0.323 (0.032) =	**0.313 (0.025) +**
ccrime	0.156 (0.006)	0.152 (0.009) =	0.152 (0.007) =	0.1504 (0.008) =	0.153 (0.007) =	0.1504 (0.007) =
bioava	33.18 (3.71)	31.77 (10.08) =	31.50 (3.20) =	30.18 (1.75) +	31.78 (3.39) =	***28.78 (1.83)*** **+**

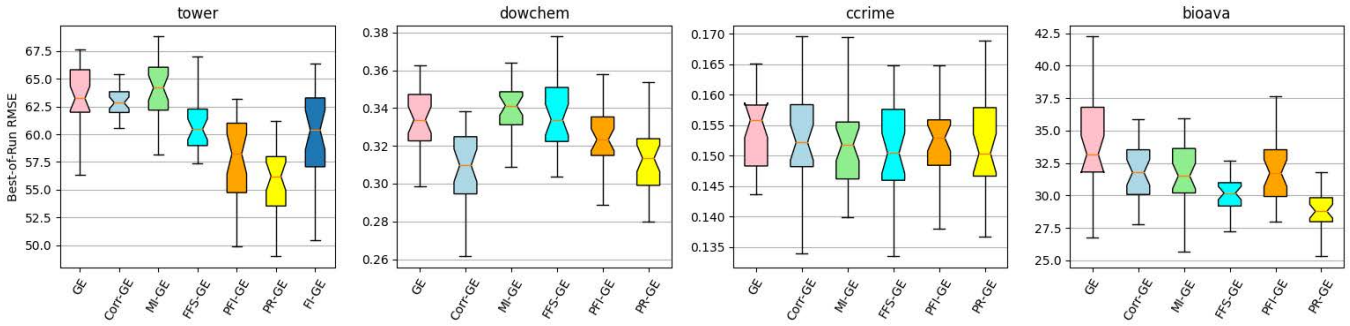


Figure 4: Accuracy comparison of PR-GE with other feature selection based GE methods

TRFs, there is lot of feature diversity in the population. Since there are only arithmetic operators to bind terminals in the expression, the lack of appropriate function primitives lead GE to construct solutions which contain several AFs.

In Exp 2.6 we made the problem even harder by adding a uniformly distributed random noise to the input variables. The noise interval was [-0.3, 0.3] which is within 5% of the input range of TRFs. As before, the TRFs were randomly spread within the AFs. The corresponding PR-plot in Figure 3(d) indicate that PRFS could easily select the TRFs.

5.2 Impact on Generalization

A summary of results on the generalization performance of using various feature selection approaches is presented in Table 5. The second column mentions the baseline results with GE without feature selection. The rest of the five columns list results with each X-GE method. For each problem, the median and standard deviation (in parenthesis) of the best test scores across 30 runs is given. Statistical significance tests were conducted at 95% confidence level using 2-tailed Mann-Whitney U-test. Two iterations of pair-wise statistical comparisons were performed:

(1) GE baseline with each X-GE method. The symbol at the right end indicate outcome of statistical comparison with the baseline. The symbol '+' indicates the results are significantly better than the baseline. In case of '=', the results are comparable as statistical significance was not established.
(2) PR-GE with the rest of X-GE approaches. Where PR-GE is significantly better than all the others, the figures are shown in bold and italics, as in case of `tower` and `bioava`. In case of `dowchem`, PR-GE test score is shown only in bold (not italic) because it is significantly better than all but Corr-GE. In case of `ccrime` PR-GE result is not significantly better than the rest, therefore it is shown in regular font.

Figure 4 presents the results in box plots corresponding to each dataset. It is evident that feature selection, in most cases, irrespective of the feature selection approach, results in improved generalization performance with GE. Except for few instances, where

Exp#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	- - -
Features (outliers)	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	X31	- - -
	X34	X34	X34	X48	X34	X34	X34	X48	X48	X48	X48	X48	X48	X48	X48	X48	X34	X48	X48	X48	X48	X34	X48	X48	X48	X34	X34	X48	X48	X48	- - -
	X48	X48	X48	X34	X48	X48	X48	X34	X34	X34	X34	X34	X34	X34	X34	X34	X48	X34	X34	X34	X34	X48	X34	X34	X34	X48	X48	X34	X34	X34	- - -
	X30	X30	X46	X30	X36	X46	X46	X30	X30	X36	X30	X30	X30	X36	X30	X30	X46	X46	X36	X46	X36	X36	X30	X30	X30	X36	X30	X46	X30	X36	- - -
	X46	X36	X30	X46	X30	X30	X30	X45	X46	X30	X46	X46	X36	X30	X36	X46	X30	X36	X45	X45	X46	X30	X36	X46	X46	X30	X36	X36	X46	X45	- - -
	X45	X45	X45	X36	X46	X36	X36	X36	X36	X45	X45	X36	X45	X45	X46	X45	X45	X30	X46	X30	X30	X46	X46	X45	X36	X46	X46	X45	X45	X46	- - -
	X36	X46	X36	X45	X45	X45	X45	X46	X45	X46	X36	X45	X46	X5	X45	X36	X36	X45	X30	X36	X45	X45	X45	X36	X45	X45	X45	X30	X36	X30	- - -
	X15	X15			X15	X5			X15	X5	X15	X15	X15	X46	X15	X15	X15	X15	X15	X15	X14	X5	X5	X15	X5		X5	X15	X5	X15	- - -
		X5			X5	X15							X5	X14			X14	X5		X5				X5	X15		X15			X5	- - -
														X15						X14					X14						- - -

Figure 5: Features selected by PRFS (shown in color coding) in first 30 experiments out of 100 experiments on dowchem

the performance is slightly degraded, the test scores are either significantly or slightly better.

For each dataset, we compared PR-GE with four other X-GE methods, except for tower where we used a fifth method as well. This is because the tower dataset was initially studied by Smit et. al. [34] for selecting features using their approach of Fitness Inheritance (FI). Vladislavleva et al. [35] used the dataset with selected features. We used the same set of selected features with GE, which is labelled as FI-GE in the left-most plot in Figure 4. For this dataset, PR-GE significantly improves generalization performance in GE. Similarly, for the bioava dataset, PR-GE test results were significantly better than all other methods. However, for the ccrime dataset, results from all X-GE methods were comparable. In all pair-wise comparisons, no method was significantly better than the other on this data set.

5.3 Consistency of Production Ranking

Our production ranking algorithm is a stochastic process since it is based on an underlying stochastic search which, in our case, is a genetic algorithm. The GA explores the space of variable length binary strings which is mapped to the solution space by GE. Since the exact candidate solutions generated, evaluated, and subsequently analyzed for production ranking would not be the same in each trial, the ranks assigned to each production may vary. However, since the evolution learns which solution elements (or productions) are suitable for the problem at hand, we expect to achieve similar ranks in every experimental run. We therefore hypothesized that production ranking scores in PRFS would be fairly consistent. The direct consequence would be the selection of similar sets of features among any two independent ranking experiments.

In order to test our hypothesis, we conducted 100 PRFS experiments on the dowchem dataset. In each experiment, we conducted 30 independent trials with the parameters described in section 4.3. The outcome is shown in Figure 5. The selected features (for which the corresponding production ranks are outliers) in each experiment are listed column-wise. Each feature is color-coded to better visualize the distribution and similarity across experiments. Due to space limitations, only first 30 experiments are shown.

In all 100 experiments, the number of selected features vary from 7 to 10. To obtain a numerical measure of consistency or similarity of selected features across experiments, we used the *Jaccard Coefficient* [15]. It measures the similarity between two sets and is defined as a ratio of the size of the intersection to the size of the union. In the equation 6 below, $J(A, B)$ is the Jaccard coefficient between sets A and B. Note that in our case, A and B are the feature sets from two distinct experiments.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \qquad \text{where } 0 \leqslant J(A,B) \leqslant 1 \tag{6}$$

For ith experiment, the mean Jaccard coefficient $\bar{J}(A_i)$ was computed which is the mean of $J(A_i, B_j)$ with the rest of experiments (99 in our case) excluding self-similarity computation:

$$\bar{J}(A_i) = \frac{1}{k-1} \sum_{j=1, j\neq i}^{k} J(A_i, B_j) \tag{7}$$

$$\bar{\bar{J}}(Exp_1^k) = \frac{1}{k} \sum_{i=1}^{k} \bar{J}(A_i) \tag{8}$$

Equation (8) gives the Jaccard coefficient for all k experiments. In our case k=100. The Jaccard coefficient across all 100 experiments was computed to be 0.844, which indicates fairly high similarity and thus a high consistency of feature selection through PRFS. It is interesting to note that when top 7 features (lowest cardinality of outliers) are considered in each experiment, the similarity score became 0.995 which indicate very high consistency.

6 CONCLUSION

We presented a grammar-based feature selection technique. Through production ranking and a grammar pruning mechanism, this can be easily plugged in to GE. Through extensive experimentation, it has been found that production-ranking based feature selection (PRFS) can identify relevant features, which is particularly useful for high-dimensional symbolic regression. PRFS was tested on a set of problems varying from low to high dimensionality problems and we demonstrated that it has the ability to significantly enhance generalization performance in symbolic regression.

These promising results have led us to explore further, and several detailed investigations, improvements and extensions are underway. We are applying PRFS to a larger problem set including classification problems. In addition, improved ranking strategies are being devised; currently, the production ranking scheme assigns ranks by equally weighing all variables in the solution expression. We are in a process to employ more opportunistic production ranks to assess marginal contributions of each feature in the solution using approaches such as variants of Shapely-Value, without adding much computational overhead.

ACKNOWLEDGMENTS

This work was supported with the financial support of the Science Foundation Ireland grant 16/IA/4605 to the project Automatic Design of Digital Circuits (ADDC) at University of Limerick, Ireland.

REFERENCES

[1] Baligh Al-Helali, Qi Chen, Bing Xue, and Mengjie Zhang. 2020. Genetic Programming with Noise Sensitivity for Imputation Predictor Selection in Symbolic Regression with Incomplete Data, Yaochu Jin (Ed.). *IEEE Congress on Evolutionary Computation (CEC)*, 1–8. https://doi.org/doi:10.1109/CEC48606.2020.9185526

[2] Muhammad Sarmad Ali, Meghana Kshirsagar, Enrique Naredo, and Conor Ryan. 2021. AutoGE: A Tool for Estimation of Grammatical Evolution Models. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence*. SCITEPRESS - Science and Technology Publications, 1274–1281. https://doi.org/10.5220/0010393012741281

[3] Muhammad Sarmad Ali, Meghana Kshirsagar, Enrique Naredo, and Conor Ryan. 2021. Towards Automatic Grammatical Evolution for Real-world Symbolic Regression. In *Proceedings of the 13th International Conference on Evolutionary Computation Theory and Applications (ECTA'21)*. SCITEPRESS - Science and Technology Publications, 68–78. https://doi.org/10.5220/0010691500003063

[4] Francesco Archetti, Stefano Lanzeni, Enza Messina, and Leonardo Vanneschi. 2006. Genetic programming for human oral bioavailability of drugs. In *Proceedings of the 8th annual conference on Genetic and Evolutionary Computation*, Vol. 1. ACM Press, 255–262. https://doi.org/doi:10.1145/1143997.1144042

[5] R Muhammad Atif Azad and Conor Ryan. 2005. An Examination of Simultaneous Evolution of Grammars and Solutions. In *Genetic Programming Theory and Practice III*, Tina Yu, Rick L Riolo, and Bill Worzel (Eds.). Genetic Programming, Vol. 9. Kluwer, Ann Arbor, Chapter 10, 141–158. https://doi.org/doi:10.1007/0-387-28111-8_10

[6] Avrim L. Blum and Pat Langley. 1997. Selection of relevant features and examples in machine learning. *Artificial Intelligence* 97, 1-2 (dec 1997), 245–271. https://doi.org/10.1016/S0004-3702(97)00063-5

[7] Qi Chen, Bing Xue, Ben Niu, and Mengjie Zhang. 2016. Improving generalisation of genetic programming for high-dimensional symbolic regression with feature selection. *2016 IEEE Congress on Evolutionary Computation, CEC 2016* (2016), 3793–3800. https://doi.org/10.1109/CEC.2016.7744270

[8] Qi Chen, Mengjie Zhang, and Bing Xue. 2017. Feature Selection to Improve Generalisation of Genetic Programming for High-Dimensional Symbolic Regression. *IEEE Transactions on Evolutionary Computation* 21, 5 (oct 2017), 792–806. https://doi.org/doi:10.1109/TEVC.2017.2683489

[9] Grant Dick, Aysha P. Rimoni, and Peter A. Whigham. 2015. A re-examination of the use of genetic programming on the oral bioavailability problem. *GECCO 2015 - Proceedings of the 2015 Genetic and Evolutionary Computation Conference* (2015), 1015–1022. https://doi.org/10.1145/2739480.2754771

[10] Dimitris Gavrilis, Ioannis G. Tsoulos, and Evangelos Dermatas. 2008. Selecting and Constructing Features Using Grammatical Evolution. *Pattern Recogn. Lett.* 29, 9 (jul 2008), 1358–1365. https://doi.org/10.1016/j.patrec.2008.02.007

[11] Isabelle Guyon and André Elisseeff. 2003. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research* 3, null (mar 2003), 1157–1182.

[12] Maarten Keijzer. 2003. Improving Symbolic Regression with Interval Arithmetic and Linear Scaling. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 2610. 70–82. https://doi.org/10.1007/3-540-36599-0_7

[13] John R Koza. 1993. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press.

[14] Max Kuhn and Kjell Johnson. 2013. *Applied Predictive Modeling*. Springer New York, New York, NY. https://doi.org/10.1007/978-1-4614-6849-3

[15] Michael Levandowsky and David Winter. 1971. Distance between Sets. *Nature* 234 (11 1971), 34–35. Issue 5323. https://doi.org/10.1038/234034a0

[16] Nuno Lourenco, Filipe Assuncao, Francisco B Pereira, Ernesto Costa, and Penousal Machado. 2018. Structured Grammatical Evolution: A Dynamic Approach. In *Handbook of Grammatical Evolution*, Conor Ryan, Michael O'Neill, and J J Collins (Eds.). Springer, Chapter 6, 137–161. https://doi.org/doi:10.1007/978-3-319-78717-6_6

[17] Sean Luke and Liviu Panait. 2002. Is The Perfect The Enemy Of The Good?. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. Morgan Kaufmann Publishers, New York, 820–828.

[18] Eric Medvet. 2017. Hierarchical grammatical evolution. *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (2017).

[19] Jessica Megane, Nuno Lourenco, and Penousal Machado. 2021. Probabilistic Grammatical Evolution. In *EuroGP 2021: Proceedings of the 24th European Conference on Genetic Programming (LNCS, Vol. 12691)*, Ting Hu, Nuno Lourenco, and Eric Medvet (Eds.). Springer Verlag, Virtual Event, 198–213. https://doi.org/10.1007/978-3-030-72812-0_13

[20] Mariana Monteiro, Nuno Lourenço, and Francisco B. Pereira. 2021. FERMAT: Feature Engineering with Grammatical Evolution. In *Progress in Artificial Intelligence*, Goreti Marreiros, Francisco S. Melo, Nuno Lau, Henrique Lopes Cardoso, and Luís Paulo Reis (Eds.). Springer International Publishing, Cham, 239–251.

[21] Alberto Moraglio, James McDermott, and Michael O'Neill. 2018. Geometric Semantic Grammatical Evolution. , 163-188 pages. https://doi.org/doi:10.1007/978-3-319-78717-6_7

[22] Eoin Murphy, Erik Hemberg, Miguel Nicolau, Michael O'Neill, and Anthony Brabazon. 2012. Grammar Bias and Initialisation in Grammar Based Genetic Programming. In *Proceedings of the 15th European Conference on Genetic Programming, EuroGP 2012 (LNCS, Vol. 7244)*. Springer Verlag, Malaga, Spain, 85–96. https://doi.org/doi:10.1007/978-3-642-29139-5_8

[23] Miguel Nicolau and Alexandros Agapitos. 2018. Understanding Grammatical Evolution: Grammar Design. In *Handbook of Grammatical Evolution*, Conor Ryan, Michael O'Neill, and J J Collins (Eds.). Springer International Publishing, Cham, 23–53. https://doi.org/10.1007/978-3-319-78717-6_2

[24] Miguel Nicolau and Alexandros Agapitos. 2021. Choosing function sets with better generalisation performance for symbolic regression models. *Genetic Programming and Evolvable Machines* 22, 1 (mar 2021), 73–100. https://doi.org/doi:10.1007/s10710-020-09391-4

[25] Michael O'Neill, Anthony Brabazon, Miguel Nicolau, Sean Mc Garraghy, and Peter Keenan. 2004. pi Grammatical Evolution. *Genetic and Evolutionary Computation – GECCO-2004, Part II* 3103, 617–629. https://doi.org/doi:10.1007/b98645

[26] Michael O'Neill, Leonardo Vanneschi, Steven Gustafson, and Wolfgang Banzhaf. 2010. Open issues in genetic programming. *Genetic Programming and Evolvable Machines* 11, 3/4 (sep 2010), 339–363. https://doi.org/doi:10.1007/s10710-010-9113-2

[27] Alfonso Ortega, Marina de la Cruz, and Manuel Alfonseca. 2007. Christiansen grammar evolution: Grammatical evolution with semantics. *IEEE Transactions on Evolutionary Computation* 11, 1 (2007), 77–90. https://doi.org/10.1109/TEVC.2006.880327

[28] James Vincent Patten and Conor Ryan. 2015. Attributed Grammatical Evolution using Shared Memory Spaces and Dynamically Typed Semantic Function Specification. In *18th European Conference on Genetic Programming (LNCS, Vol. 9025)*. Springer, Copenhagen, 105–112. https://doi.org/10.1007/978-3-319-16501-1_9

[29] Conor Ryan and R Muhammad Atif Azad. 2003. Sensible Initialisation in Grammatical Evolution, Alwyn M Barry (Ed.). *{GECCO 2003}: Proceedings of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference*, 142–145.

[30] Conor Ryan, JJ Collins, and Michael O'Neill. 1998. Grammatical evolution: Evolving programs for an arbitrary language. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, W. Banzhaf, R. Poli, M. Schoenauer, and T.C. Fogarty (Eds.). Vol. 1391. Springer, Berlin, Heidelberg, 83–96. https://doi.org/10.1007/BFb0055930

[31] Conor Ryan, Michael O'Neill, and JJ Collins. 2018. Introduction to 20 Years of Grammatical Evolution. In *Handbook of Grammatical Evolution*. Springer International Publishing, 1–21. https://doi.org/10.1007/978-3-319-78717-6_1

[32] Ashish Sen and Muni Srivastava. 1990. *Regression Analysis*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-25092-1

[33] Anthony Mihirana De Silva, Farzad Noorian, Richard I.A. Davis, and Philip H.W. Leong. 2013. A Hybrid Feature Selection and Generation Algorithm for Electricity Load Prediction Using Grammatical Evolution. In *2013 12th International Conference on Machine Learning and Applications*, Vol. 2. 211–217. https://doi.org/10.1109/ICMLA.2013.125

[34] Guido Smits, Arthur Kordon, Katherine Vladislavleva, Elsa Jordaan, and Mark Kotanchek. 2005. Variable Selection in Industrial Datasets using Pareto Genetic Programming. In *Genetic Programming Theory and Practice III*, Tina Yu, Rick L Riolo, and Bill Worzel (Eds.). Genetic Programming, Vol. 9. Springer, Chapter 6, 79–92. https://doi.org/doi:10.1007/0-387-28111-8_6

[35] Ekaterina J Vladislavleva, Guido F Smits, and Dick den Hertog. 2009. Order of Nonlinearity as a Complexity Measure for Models Generated by Symbolic Regression via Pareto Genetic Programming. *IEEE Transactions on Evolutionary Computation* 13, 2 (Apr 2009), 333–349. https://doi.org/doi:10.1109/TEVC.2008.926486

[36] David R. White, James McDermott, Mauro Castelli, Luca Manzoni, Brian W. Goldman, Gabriel Kronberger, Wojciech Jaśkowski, Una May O'Reilly, and Sean Luke. 2013. Better GP benchmarks: Community survey results and proposals. *Genetic Programming and Evolvable Machines* 14, 1 (2013), 3–29. https://doi.org/10.1007/s10710-012-9177-2

[37] T Wong and P Yeh. 2020. Reliable Accuracy Estimates from k-Fold Cross Validation. *IEEE Transactions on Knowledge and Data Engineering* 32, 8 (aug 2020), 1586–1594. https://doi.org/10.1109/TKDE.2019.2912815

[38] Bing Xue, Mengjie Zhang, Will N. Browne, and Xin Yao. 2016. A Survey on Evolutionary Computation Approaches to Feature Selection. *IEEE Transactions on Evolutionary Computation* 20, 4 (2016), 606–626. https://doi.org/10.1109/TEVC.2015.2504420