Noname manuscript No.
(will be inserted by the editor)

A New Imputation Method Based on Genetic Programming and Weighted KNN for Symbolic Regression with Incomplete Data

Baligh Al-Helali · Qi Chen · Bing Xue · Mengjie Zhang

the date of receipt and acceptance should be inserted later

Abstract Incompleteness is one of the problematic data quality challenges in real-world machine learning tasks. A large number of studies have been conducted for addressing this challenge. However, most of the existing studies focus on the classification task and only a limited number of studies for symbolic regression with missing values exist . In this work, a new imputation method for symbolic regression with incomplete data is proposed. The method aims to improve both the effectiveness and efficiency of imputing missing values for symbolic regression. This method is based on genetic programming (GP) and weighted K-nearest neighbors (KNN). It constructs GP-based models using other available features to predict the missing values of incomplete features. The instances used for constructing such models are selected using weighted KNN. The experimental results on real-world data sets show that the proposed method outperforms a number of state-of-the-art methods with respect to the imputation accuracy, the symbolic regression performance, and the imputation time.

Keywords Symbolic Regression · Genetic programming · Incomplete Data · KNN · Imputation.

1 Introduction

Symbolic regression (SR) is the process of constructing mathematical models that express the output variable in terms of input variables [18]. While traditional regression techniques work by optimizing the parameters for a prespecified model, symbolic regression discovers the model with its parameters without any assumptions. This makes symbolic regression applicable to many real-world applications, especially when dealing with multivariate data from

School of Engineering and Computer Science
Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand
Email: E-mail: {baligh.al-helali, Qi.Chen, Bing.Xue, Mengjie.Zhang}@ecs.vuw.ac.nz

unknown systems such as materials science [45]. Symbolic regression is also utilized for many machine learning tasks such as reinforcement learning [20]. However, in reality, the given data can be incomplete. According to [19], a poor generalization of regression models can often be a result of incomplete data. Moreover, incomplete data sets are difficult to be handled directly by most regression methods.

Many real-world regression data sets are annotated as having missing values according to data repositories such as UCI [8] and OpenML [41]. Missing values are due to many reasons such as bad-designed questionnaires and failures in data collection devices. Methods for handling missing values should be chosen carefully as this may affect the overall performance of the learning process.

There are many methods that have been proposed to address the data incompleteness including deletion-based methods, model-based methods, implicit learning-based methods, and imputation-based methods [12]. For the imputation methods, the main idea is to provide estimate values for the missing ones [9]. The imputation approach is widely adopted as it enables the use of various learning methods after producing complete data [37]. However, powerful imputation methods are typically expensive, since they are suitable for batch imputation where chunks of training instances are required to impute a single instance [37].

The existing research on dealing with missing values mainly focuses on classification tasks. Many imputation methods are developed for classification with missing values [12]. In symbolic regression research, only a few studies consider imputation for symbolic regression [1,28]. Moreover, they have some limitations. In [28], missing values are simply replaced with corresponding feature values from other instances. The method presented in [1] reuses the training data to build imputation models for every missing value in the test data, which makes it computationally expensive. Therefore, more effort is needed for symbolic regression with incomplete data.

Genetic Programming (GP) is a popular evolutionary algorithm that generates computer programs for performing a user-defined task automatically [17]. It starts from a high-level definition of the problem and creates a population of random programs and then refines them progressively using genetic operators (e.g. crossover and mutation) and selection strategies until obtaining a satisfactory solution. GP-based imputation (GPI) is adopted successfully for classification with missing values in [38–40]. The main idea is to consider each feature having missing values as the target variable while using other features as predictive variables. Instances with complete corresponding feature values are used to build the regression functions and these functions are then used to predict the missing values. This method has the advantage of not requiring any presumptions. However, it performs the regression using all instances regardless of their similarity. GPI might use some instances that are irrelevant to the instance to be imputed. On the other hand, K-nearest neighbour (KNN) imputation replaces the missing value with the average of the K closest instances. Although KNN clearly takes the instance-based relevance into account, it ignores the feature-based relevance.

Recently, we proposed a hybrid imputation method for symbolic regression with missing values in [1]. This method is called GP-KNN, which combines GPI and KNN. Such combination utilizes both the GPI feature-based predictability and the KNN instance-based similarity to impute missing values. GPI is used to build regression-based imputation models which are learned from similar instances selected by KNN. GP-KNN has promising imputation performance, and it outperforms popular imputation methods such as classification and regression trees (CART), random forest (RF), and KNN, in many cases. However, GP-KNN builds new imputation models when imputing new incomplete instances, which is computationally expensive. Moreover, GP-KNN requires the use of the training data during the test phase, which means more space complexity.

In this work, we will develop a new method based on the GP-KNN imputation method to achieve more efficient and effective imputation when performing symbolic regression. To improve the effectiveness, instead of the simple KNN method, the weighted KNN is used. Weighted KNN increases the contributions of the instances which are close to the incomplete instances when building the imputation models. Meanwhile, the efficiency is enhanced by avoiding the requirement of building new imputation models when imputing new unseen instances. Specifically, the objectives of this work include:

1. developing a hybrid imputation method by combining both feature-based prediction using GP and instance-based similarity using weighted KNN;
2. designing a training process that detects and constructs different missingness knowledge components such as missingness patterns, feature-based prediction models, and instance-based similarity references;
3. providing an efficient test process that can utilize the outputs of the training process to impute unseen instances without the expensive requirement of building new models; and
4. investigating whether the proposed method can outperform state-of-the-art imputation methods w.r.t the imputation accuracy, the symbolic regression performance, and the imputation time.

2 Related Work

This section presents the related work. This includes research on symbolic regression, incomplete data, and evolutionary computation-based imputation methods.

2.1 Symbolic Regression

Symbolic regression is a machine learning field whose task is to construct a mathematical model that best fits a given data set. Many statistical techniques have been utilized for addressing regression tasks. However, some presumptions might be required for solving regression tasks, such as linear relationship, no auto-correlation, and multivariate normality. For symbolic regression, less pre-assumptions are required and it is able to learn the structure of the model and the coefficients simultaneously [18].

There are several evolutionary computation (EC) techniques that have been used for symbolic regression. Genetic Programming (GP) has been typically applied for solving symbolic regression problems [17]. In [2], a continuous neural encoding approach is proposed to improve conventional linear representation in genetic programming for symbolic regression. A model-based GP approach is proposed for symbolic regression of small expressions in [43]. They use a gene-pool optimal mixing evolutionary algorithm (GOMEA) model-based EA framework for symbolic regression and called it GP-GOMEA. GP-based methods do not need to impose restrictions on how the structure of solutions should be. Moreover, it is possible to get accurate results without upfront analytical knowledge. However, GP requires a large search space which means a long time is needed to find a suitable model [7].

Grammatical evolution (GE) is used for symbolic regression in [26]. The application of GE requires employing a search strategy (e.g. simulated annealing, hill climbing, random search, and genetic algorithms) and the performance relies on the search strategy. Another EC-based method called artificial immune system (AIS) is used for symbolic regression [15]. AIS is similar to GP as the programs are represented as trees. However, components of the evolutionary process of GP are translated into the immune metaphor. The experimental results show that the AIS method typically converges slightly quicker than the standard GP method. However, the size of trees evolved using AIS is larger than that of GP.

In [47], analytic programming (AP) is used for symbolic regression. It is based on GP and Hilbert spaces aiming to address the representation of a symbolic model. The reported comparisons show that AP is equivalent to GP on the considered tasks. However, it needs to be expanded with more comparative analysis focusing on the complexity of the addressed problems.

A non-EC method called Fast Function Extraction (FFX) is proposed for symbolic regression in [24]. FFX uses a pathwise regularized learning technique for pruning a large set of candidate basis functions down to a smaller compact set of models. In [5], another non-EC real-time algorithm for symbolic regression is proposed. This algorithm is called Elite Bases Regression (EBR) and it works by generating a set of candidate basis functions coded with parse-matrix by specific mapping rules. Some of the elite bases are updated iteratively based on the correlation with the target model. The regression model is then spanned by the elite bases. A comparison between EBR and FFX is conducted and the reported results indicate that EBR is able to solve symbolic regression problems more effectively. However, several control parameters should be tuned in the EBR method. In [16], a deterministic symbolic regression algorithm is proposed using a context-free grammar. It utilises non-linear least squares local optimisation to produce the symbolic regression models. However, structural restrictions are imposed to guarantee finite enumeration of the possible models.

GP has a flexible representation ability and a symbolic nature of its solutions, which make it the most commonly used approach for symbolic regression [6]. Therefore, GP approach is adopted in this work in order to make the proposed method more usable in the symbolic regression community.

2.2 Incomplete Data

There are different categorise of missing values including missing completely at random (MCAR), missing not at random (MNAR), and missing at random (MAR) [9]. In MCAR missingness, the missing values are a random subset of the given data set with a completely random reason. In this kind of missingness, the missingness probability of a value is not related to any other value. When missing values are MCAR, most handling missing techniques give unbiased results [9]. This is because no gain can be guaranteed by analyzing the available data to estimate associations with missing values. For MNAR missingness, the probability that a value is missing is related to non-observed information [31]. Due to the lack of valuable information, there is no universal method to handle this type of missingness. Usually, missing values are neither MCAR nor MNAR [9]. Instead, the probability of missingness is commonly related to information that is present, i.e., missingness depends on other existing values. This type of missingness is called missing at random (MAR). This term is confusing, however, it can be justified as missing values may indeed be considered random conditional on other values [31]. When the missingness is MAR, most statistical techniques for handling missing values give biased results [31]. However, more sophisticated methods (e.g. regression-based multiple imputation) may give good results [9].

Let X be a variable represents the given data, where X_m by a variable representing the missing data and X_o be a variable refers to complete;y observed data. Now let R be a variable represents the missingness, then the different kinds of missingness mechanisms can be expressed using conditional probability as follows [34]. MCAR: $P(R|X) = P(R)$, i.e. probability of missingness does not depend on observed X_o. MAR: $P(R|X) = P(R|X_o)$, i.e. probability of missingness depends on X_o. MNAR: $P(R|X) = P(R|X_m)$, i.e. probability of missingness depends on unobserved X_m . In this work, we will focus on MAR as it is more suitable for evaluating the performance of data imputation.

Methods for handling missing values can be categorized into four different approaches [12]. The first approach is to delete either incomplete instances or incomplete features and learn from the complete data portion only. This approach is simple but it might cause a loss of informative data. Another approach is to impute the missing values and then apply the learning algorithm on a complete data set after imputation. This approach enables the use of different learning methods but usually, it is time-consuming. Model-based procedures are used to model incomplete data distribution such as expectationmaximization (EM). This approach requires assumptions on the joint distribution of the variables. Some machine learning algorithms can directly deal with incomplete instances without explicitly estimating the missing values such as decision trees, fuzzy approaches, and ensemble methods.

For imputation, there are two main types of methods: single imputation and multiple imputation. Single imputation directly replaces the missing values using the estimated ones based on observed characteristics. The common problem with single imputation is that it underestimates the variance and ignores uncertainty. Multiple imputation addresses this problem by considering both within-imputation uncertainty and between-imputation uncertainty [36]. In multiple imputation, multiple complete copies of the data are produced. Each is an independent estimate for the missing values. Then, a statistical analysis is conducted on the imputed data sets separately and the results are combined to get an estimated value [36].

2.3 Evolutionary Computation-based Imputation

Several EC-based imputation methods have been developed for imputing missing values. In [11], a genetic algorithm (GA) is used to impute missing values in multivariate data. In [21], a multi-objective GA is used for imputation, which can deal with numerical and nominal features. It optimizes both the classification accuracy and the imputation predictability. In [30], a GA is used to impute missing values in discrete features. In [27,22], GA-based imputation is also used for classification with missing values.

In [13], two imputation methods are proposed by combining particle swarm optimization (PSO), auto-associative extreme learning machine, and evolving clustering method. In [33], a hybrid imputation method combining PSO and Fuzzy C-Means (FCM) is presented. The FCM is used to extract similar complete instances. Then, PSO is applied to optimize the instances based on complete information. In [32], a Fuzzy Swarm imputation method is proposed. A fuzzy-based clustering of the complete instances is firstly applied as a preprocessing stage to fill in the missing values and then PSO is used to optimize the imputation.

GP-based imputation methods have been successfully used for classification. A GP-based multiple imputation method that utilizes the symbolic regression prediction ability of GP to estimate missing values in classification data sets is proposed in [38]. To impute incomplete test instances, the whole data (including the training set) are used to build regression models whose target is the feature that contains missing values. The model with the smallest fitness value is then applied to estimate the missing values. This process is repeated for each incomplete feature. In [39], the GP-based imputation is modified to have two separate processes; the training process and the test process. In the training process, imputation regression models are constructed using the training data set. The test process is then performed to impute single test instances by applying the constructed imputation regression models. In [40], multiple imputation and GP are combined to evolve classifiers on data with missing values. Common patterns of missing values are firstly extracted and GP is then used to construct a classifier for each pattern.

2.4 Symbolic Regression with Missing Data

In [44], missing values in certain ranges of the feature space are considered to be causing an imbalanced data problem. To handle this problem, a framework for automatic weighting of data points is suggested, which considers the relative importance of the points using four importance weighting schemes according to the proximity, remoteness, surrounding, and nonlinear deviation from nearest neighbors. The methods are used to balance synthetic data sets drawn from mathematical functions. The applicability of these methods is not validated on real-world incomplete data.

A hybrid imputation method called GP-KNN is presented in [1]. This method is employed for symbolic regression with missing values. The main idea of this method is to combine the regression-based imputation of GP and the instance-based selection of KNN. This method is evaluated using two measures; the imputation accuracy and the symbolic regression error. The experimental results show that GP-KNN outperforms state-of-the-art imputation methods on real-world data sets. However, the main limitation of the GP-KNN method is the time complexity of the imputation process. This drawback is due to the need for building new imputation models for each missing value in the test data using the training data.

3 Weighted KNN-GP Imputation for Symbolic Regression

In this section, the details of the proposed method are presented in terms of the overall approach, the training process, and the test process.

3.1 The Overall Approach

In many imputation methods, e.g. multivariate imputation by chained equations (MICE), batch imputation is performed by using the existing values to impute the missing values in a given set of instances [37]. However, such a process is not applicable for imputing a single test instance. Hence, the training data are usually needed when imputing new test instances. This requirement increases the complexity in terms of both space and time, especially for tasks with large data sets.

In this work, the imputation method is designed to be suitable for imputing single test instances without the need to reuse the training data. The training set is used to construct the imputation models that are applied later to impute the missing values in new unseen data. These models are generated by GP using instances selected by weighted KNN. Therefore, this method is called

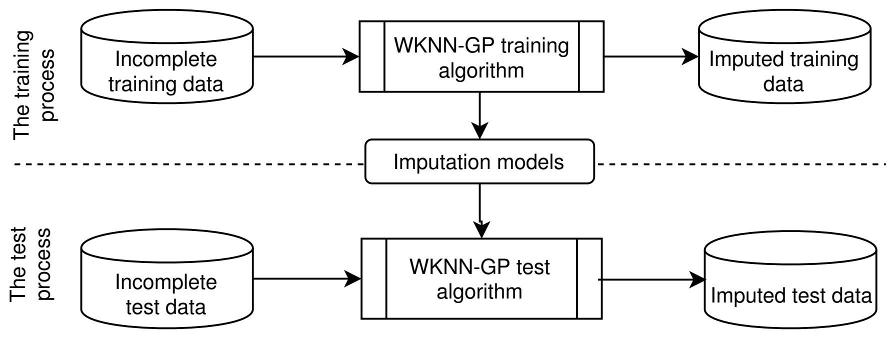

Fig. 1: The overall approach of using the proposed WKNN-GP method for imputation.

weighted KNN-GP (WKNN-GP). The WKNN-GP method has two processes: the WKNN-GP training process and the WKNN-GP test process.

Fig. 1 shows the overall approach of using the proposed method for imputation. As it shows, the input of the WKNN-GP training process is an incomplete training data set while the outputs are sets of imputation models and a complete imputed training data set. After the construction of imputation models, the WKNN-GP test process uses these models to impute the missing values in the test data set and produces a complete imputed test data set.

3.2 The WKNN-GP Training Process

3.2.1 The main idea

The training process aims to provide imputation models for the missing values in a given incomplete data set. The process of creating these models has two main steps. The first one is to utilize weighted KNN to extract K instances nearest to the incomplete instance. The second step employs GP imputation to build prediction models using the retrieved K instances. These models are constructed by fitting the features with missing values (i.e. the prediction target) and involving the weights of the K instances in the GP fitness function. The similarity between instances is measured using the normalized Euclidean distance given in Equation (1).

$$\begin{aligned} distance(\mathbf{p},\mathbf{q}) &= \sqrt{\frac{(q_1-p_1)^2}{r_1^2}+\frac{(q_2-p_2)^2}{r_2^2}+\cdots+\frac{(q_n-p_n)^2}{r_n^2}} \\ &= \sqrt{\sum_{i=1}^{n}\frac{(q_i-p_i)^2}{r_i^2}}, \end{aligned} \tag{1}$$

where **p**, **q** are real-valued n-dimensional instances, and r_i is the range of the i^{th} feature.

The use of weighted KNN gives more importance to the instances that are close to the incomplete ones. The closer the instance, the higher contribution. On the other hand, the GP technique is used to generate imputation models due to its successful application for classification with incomplete data [38,39].

Example 1 To clarify the high-level steps of the proposed method, a simple example is given. This example is also helpful to explain the detailed algorithm later. Considering the incomplete data set shown in Table 1 (the missing values are represented as ?), two main steps are applied to perform WKNN-GP imputation for the missing value at $(i, j) = (1, 2)$, i.e. the value of the feature F_2 in the instance I_1.

Table 1: An incomplete data set.

Instance	F_1	F_2	F_3	F_4	T: regression target
I_1	300	**?**	1300	?	33.7
I_2	310	10	1200	12	35.87
I_3	200	40	1700	?	30.73
I_4	?	30	1100	15	39.17
I_5	320	?	1600	13	39.11
I_6	300	60	1400	17	35.01
I_7	200	40	1200	11	29.77
I_8	290	80	1400	11	31.1
I_9	?	?	1000	14	39.7
I_{10}	230	?	1800	?	37.3
I_{11}	290	?	1400	?	30.1
I_{12}	280	90	1500	15	31.3

Firstly, the existing values in the first instance are extracted to form a complete instance and weighted KNN is used to get its K (let $K = 2$) nearest instances. The output of this step is shown in Table 2. After that, GP is used to build imputation models using the data in Table 2 as input instances and the corresponding values of the 2^{nd} feature as target values. In other words, GP constructs imputation models for a new problem whose data set is shown in Table 3. The best constructed GP model is then used to estimate the missing value at $(1, 2)$. This model is also used to impute similar instances such as I_{11}.

Table 2: Complete sub-instances representing 2-nearest instances to the first instance.

Instance	F_1	F_3
I_2	310	1200
I_6	300	1400

Table 3: The selected 2-nearest instances with the new target variable.

Instance	F_1	F_3	Imputation target (from F_2)
I_2	310	1200	10
I_6	300	1400	60

3.2.2 The training algorithm

The pseudo-code of the training procedure is shown in Algorithm 1. It receives an incomplete training data set $\mathcal{X}$ and outputs different sets that are used to capture the imputation models for the missing values. The set G is for the constructed GP-based imputation models. R is a set of complete reference sub-instances extracted from the incomplete instances under processing. P is a set of missingness patterns representing different forms of incompleteness. The set S contains the means of the numerical features and the modes of the categorical features.

Each missing value is firstly compared with the already constructed models. If a similar model is found, the corresponding imputer is used to estimate the missing value. Otherwise, KNN is used to obtain the K nearest complete instances with their distance weights according to the normalised Euclidean distance. The obtained instances are used to build GP imputation models involving the weights of the instances in the GP fitness function. The GP imputation models are built by considering the feature containing the missing value as a regression target. Finally, the best model (i.e. the one with the best fitness value) is selected.

3.2.3 Data preparation

For each missing value at position (i, j), a missingness pattern $P_{i,j}$ is formed to reflect which features have missing values in the i^{th} instance. Non-missing values are extracted to form a complete instance $V_{i,j}$. This instance is compared with the existing reference instances in the set R that have the same missingness pattern. If there is a similar one $R_{\hat{i},j}$, the corresponding imputation model $G_{\hat{i},j}$ is used to impute $\mathcal{X}[i, j]$ directly, i.e. $\mathcal{X}^C[i, j] = G_{\hat{i},j}(V_{i,j})$.

Otherwise, new models are constructed and $V_{i,j}$ is used as a reference instance $R_{i,j}$. The features included in $P_{i,j}$ are used to get a data subset $\mathcal{X}_{i,j}^{tmp}$ after removing the instances that have missing values at the j^{th} feature. A complete data subset $\mathcal{X}_{i,j}$ is then formed by including only the complete instances from $\mathcal{X}_{i,j}^{tmp}$.

In Example 1, the missingness pattern for the missing value $\mathcal{X}[1, 2]$ is $P_{1,2} = 1010$, where the zeros refer to the missing values. From the 1^{st} instance, extract the complete instance $V_{1,2}$. As there are no previously learned imputation models, no need to compare $V_{1,2}$ with the reference set R and $V_{1,2}$ is used as a new reference instance $R_{1,2}$ ($R_{1,2} = V_{1,2}$). A data subset $\mathcal{X}_{1,2}^{tmp}$ is formed by excluding the features F_2 and F_4, and the instances I_1, I_5, I_9, I_{10}, and I_{11}. A

Algorithm 1: WKNN-GP Imputation: The Training Algorithm

```
Input  : Training data set $\mathcal{X}$ with missing values.
Output: Complete data set $\mathcal{X}^C$, $G$: GP imputation models, $R$: reference set, $P$:
        missingness patterns, and $S$: the mean and mode statistics of the training
        features.
1  Let $G = R = P = D = S = \phi$;
2  foreach missing value $\mathcal{X}[i,j]$ do
3      Obtain the corresponding missingness pattern $P_{i,j}$;
4      Extract the non-missing values from the $i^{th}$ instance to form a complete
         instance $V_{i,j}$;
5      if ($\exists P_{\hat{i},j} \in P$ s.t. $P_{i,j} = P_{\hat{i},j}$) and ($\exists R_{\hat{i},j} \in R$ and $D_{\hat{i},j} \in D$ s.t.
         $distance(R_{\hat{i},j}, V_{i,j}) \leq D_{\hat{i},j}$) then
6          $\mathcal{X}^C[i,j] \leftarrow G_{\hat{i},j}(V_{i,j})$
7      else
8          Use $V_{i,j}$ as a new reference instance $R_{i,j}$ ($R_{i,j} = V_{i,j}$) ;
9          Get a data subset $\mathcal{X}_{i,j}^{tmp}$ for the features included in $P_{i,j}$ after excluding
             the instances that have missing values at the $j^{th}$ feature ;
10         Obtain $\mathcal{X}_{i,j}$ as a complete data subset by taking the complete instances
             from $\mathcal{X}_{i,j}^{tmp}$ ;
11         $K \leftarrow min(max(|J_{i,j}|, \lfloor |I_{i,j}|/4 \rfloor), |I_{i,j}|)$, where $I_{i,j}$ and $J_{i,j}$ are the instance
             and feature indexes of $\mathcal{X}_{i,j}$, respectively;
12         $\mathcal{X}_{i,j}^K, \mathcal{D}_{i,j}^K, \mathcal{W}_{i,j}^K \leftarrow KNN(\mathcal{X}_{i,j}, R_{i,j}, K)$, where $\mathcal{X}_{i,j}^K$ contains the $K$ nearest
             instances, $\mathcal{D}_{i,j}^K$ is the corresponding distance set, and $\mathcal{W}_{i,j}^K$ is the
             corresponding distance-based weights set w.r.t the reference $R_{i,j}$;
13         Set the imputation target as $T_{i,j} \leftarrow \mathcal{X}[I_{i,j}^K, j]$, where $I_{i,j}^K$ is the instance
             indexes of $\mathcal{X}_{i,j}^K$;
14         for $g = 1$ to $N$ do
15             $G_g^{tmp} \leftarrow$ GP($\mathcal{X}_{i,j}^K$, $T_{i,j}$, $\mathcal{W}_{i,j}^K$);
16         end
17         Let $G_{\hat{g}}^{tmp}$ be the regression function with the least fitness value, i.e.
             $fitness(G_{\hat{g}}^{tmp}) \leq fitness(G_g^{tmp}), g = 1, ...N$ ;
18         Set $G_{i,j}$ to be $G_{\hat{g}}^{tmp}$ and use it to impute $\mathcal{X}[i,j]$ (i.e.
             $\mathcal{X}^C[i,j] \leftarrow G_{i,j}(R_{i,j})$) ;
19         $D_{i,j} \leftarrow \max \mathcal{D}_{i,j}^K$ ;
20         Update $G, R, D$, and $P$ by appending $G_{i,j}, R_{i,j}, D_{i,j}$, and $P_{i,j}$,
             respectively.
21     end
22     Calculate $S$ the statistics of $\mathcal{X}^C$ (the mean for the numerical features and the
         mode for the categorical features) ;
23 end
```

complete data subset $\mathcal{X}_{1,2}$ is then obtained by removing instances that have missing values from $\mathcal{X}_{1,2}^{tmp}$.

$$V_{1,2} = \begin{pmatrix} 300 & 1300 \end{pmatrix} \tag{2}$$

$$\mathcal{X}_{1,2}^{tmp} = \begin{pmatrix} 310 & 1200 \\ 200 & 1700 \\ ? & 1100 \\ 300 & 1400 \\ 200 & 1200 \\ 290 & 1400 \\ 280 & 1500 \end{pmatrix} \tag{3}$$

$$\mathcal{X}_{1,2} = \begin{pmatrix} 310 & 1200 \\ 200 & 1700 \\ 300 & 1400 \\ 200 & 1200 \\ 290 & 1400 \\ 280 & 1500 \end{pmatrix} \tag{4}$$

3.2.4 KNN settings

Rather than using all instances in $\mathcal{X}_{i,j}$ to build the imputation models, weighted KNN method is employed to acquire $\mathcal{X}_{i,j}^{K}$, which contains the K nearest instances to $R_{i,j}$ from $\mathcal{X}_{i,j}$. The corresponding distances $\mathcal{D}_{i,j}^{K}$ and distance-based weights $\mathcal{W}_{i,j}^{K}$ are also calculated.

The value of K is determined by Equation (5). This equation restricts the lower bound of K as the number of available features $|J_{i,j}|$ to avoid having fewer instances than the used features. The upper bound of K is chosen empirically as one-fourth the number of the retrieved instances $\lfloor |I_{i,j}|/4 \rfloor$. However, if these constraints can not be satisfied, i.e. in case of a small complete subset, all instances obtained are used $|I_{i,j}|$. $\mathcal{D}_{i,j}^{K}$ is calculated according to Equation (6), and $\mathcal{W}_{i,j}^{K}$ is computed using Equation (7). The distance used to measure the similarity in KNN is the normalized Euclidean (Equation (1)).

$$K = min(max(|J_{i,j}|, \lfloor |I_{i,j}|/4 \rfloor), |I_{i,j}|) \tag{5}$$

$$\mathcal{D}_{i,j}^{K}[k] = distance(\mathcal{X}_{i,j}^{K}[k], V_{i,j}), k = 1, ...K. \tag{6}$$

$$\mathcal{W}_{i,j}^{K}[k] = \frac{\mathcal{D}_{i,j}^{K}[k]}{D_{i,j}}, k = 1, ...K, \tag{7}$$

where $D_{i,j} = \max\limits_{k=1,...K} \mathcal{D}_{i,j}^{K}[k]$.

The application of the weighted KNN to get the K instances nearest to $R_{1,2}$ in Example 1 results in $\mathcal{X}_{1,2}^{K}$ (Equation (8)) and their weights are $\mathcal{W}_{1,2}^{K}$=[0.25 1]. The number of the valid features $|J_{1,2}|$ is 2 and the number of available complete instances $|I_{1,2}|$ is 6, hence $K = 2$ as $K = min(max(2, \lfloor 6/4 \rfloor), 6)$.

$$\mathcal{X}_{1,2}^{K} = \begin{pmatrix} 310 & 1200 \\ 300 & 1400 \end{pmatrix} \tag{8}$$

3.2.5 GP modeling

The subset $\mathcal{X}_{i,j}^{K}$ is then used to build N GP regression functions $\{G_g\}_{g=1}^{N}$ with the j^{th} feature as the target variable $T_{i,j}$. A weighted relative squared error (Equation (9)) is used as a fitness function to measure the quality of the individuals in the process of building the GP regression models. The value $\frac{1}{\mathcal{W}_{i,j}^{K}[k]}$ is considered when evaluating the fitness function as a distance penalty for the k^{th} instance. The closer the instance to the targeted incomplete one, the more contribution in the fitness function.

$$WRSE_{i,j}^{K} = \frac{\sum_{k=1}^{K} \frac{1}{\mathcal{W}_{i,j}^{K}[k]} (Y_{i,j}[k] - T_{i,j}[k])^2}{\sum_{k=1}^{K} (T_{i,j}[k] - \bar{T}_{i,j})^2} \tag{9}$$

where K is the number of instances, $Y_{i,j}$ is a vector of the predicted values, $T_{i,j}$ is the corresponding vector of the desired values, and $\bar{T}_{i,j}$ is the average of the desired values $T_{i,j}[k]$, $k = 1, 2, 3..., K$.

For Example 1, GP is used to build N regression models G_g^{tmp}, $g = 1, ..., N$ where $\mathcal{X}_{1,2}^{K}$ is the input and the 2nd feature of $\mathcal{X}$ is the target variable, i.e. $G_g^{tmp}(\mathcal{X}_{1,2}^{K}) \approx T_{1,2}$.

$$T_{1,2} = \begin{pmatrix} 10 \\ 60 \end{pmatrix} \tag{10}$$

Finally, the GP model $G_{\hat{g}}^{tmp}$ with the best fitness value is selected as the imputation model $G_{1,2}$ for the reference $R_{1,2}$ and it is used to estimate the missing value $\mathcal{X}[1, 2]$ as follows:

$$\mathcal{X}^{C}[1, 2] = G_{1,2}(R_{1,2}) \tag{11}$$

The instances that have the same missingness pattern as $P_{1,2} = 1010$ (i.e. I_{10} and I_{11}) are compared to the reference instance $R_{1,2}$ considering the associated distance $D_{1,2}$. Since the instance I_{11} is similar to I_1, the imputation model $G_{1,2}$ is used to impute the missing value $\mathcal{X}[11, 2]$. For the instance I_{10}, it is not close enough to use these models. Therefore, new models $P_{10,2}, R_{10,2}$, and $G_{10,2}$ are constructed for the missing value $\mathcal{X}[10, 2]$ by using the same process that is used to build the models for $\mathcal{X}[1, 2]$.

3.3 The WKNN-GP Test Process

To impute test instances, the constructed models during the training process are used as in Algorithm 2. The inputs of the test process are the learned imputation models and an incomplete test instance. This process outputs an imputed complete instance. For each missing value, if there is a similar imputation pattern, the imputation models associated with this pattern are considered for imputing this missing value. Otherwise, the missing value is replaced by the mean value if the feature is numerical or the mode value if the feature is categorical.

The process of imputing the values is done iteratively for each feature using other features as predictive inputs. For example, if there are n features, the first one with missing values is firstly imputed based on the complete ones. The remaining features are then imputed in order by involving both the complete and the already imputed features.

Algorithm 2: WKNN-GP Imputation: The Test Algorithm.

```
Input  : Test instance $V_{test}$, $R$: the instance-based reference set, $G$: the GP-based
         imputation models, $P$: the missingness pattern set, and $S$ contains the
         means and modes of the training features.
Output: Complete instance $V^C_{test}$
1  Set $V^C_{test} \leftarrow V_{test}$ ;
2  foreach missing value $V^C_{test}[j]$ do
3      Extract the non-missing values to form a complete instance $V_{test,j}$ ;
4      Obtain the corresponding missingness pattern $P_{test,j}$ ;
5      if there is a training pattern $P_{\hat{i},j} \in P$ which is similar to $P_{test,j}$ then
6          From the training reference instances that have the same missingness
             pattern $P_{\hat{i},j}$, get $R_{\hat{i},j}$ as the nearest one to $V_{test,j}$ , i.e. get $R_{\hat{i},j} \in R$ s.t.
             $distance(R_{\hat{i},j}, V_{test,j}) \leq distance(R_{i,j}, V_{test,j}), \forall i \neq \hat{\imath}$ ;
7          Set $V^C_{test}[j] \leftarrow G_{\hat{i},j}(V_{test,j})$; where $G_{\hat{i},j}$ is the corresponding GP imputer;
8      else
9          Use $S$ to estimate $V^C_{test}[j]$ by the mean if the $j^{th}$ feature is numerical or
             the mode if it is categorical.
10     end
11 end
```

Considering Example 1, let V_{test} be an incomplete test instance shown in Equation (12). The WKNN-GP imputation test process starts with the first missing value $V_{test}[2]$. To impute this value, all imputation models whose missingness patterns match $P_{test,2} = 1010$ are considered. Among the reference instances of this pattern, the nearest one to the test instance is $R_{1,2}$, hence, the GP imputation model $G_{1,2}$ is applied to impute this missing value, i.e. $V^C_{test}[2] = G_{1,2}(V_{test,2})$, where $V_{test,2} = [310\ 1400]$. A similar process is performed to impute the last missing value but with the missingness pattern $P_{test,4} = 1110$ as the previous missing value is already imputed.

$$V_{test} = \begin{pmatrix} 310 & ? & 1400 & ? \end{pmatrix} \tag{12}$$

4 Experiment Setup

In this section, the settings used for the experimental evaluations are presented. They include the benchmark data sets, the baseline methods, the parameters of the methods, and the performance evaluation methods.

4.1 Data Sets

To investigate the imputation performance of the proposed method, ten real-world data sets are used. The used data sets are shown in Table 4 and more details can be found in the data repositories UCI [8] and OpenML [41]. For each data set, the instances are split randomly in a ratio of 70:30 to form the training and testing data sets. For each pair, 30 incomplete data sets are generated using MAR missingness with different probabilities to test the ability of the proposed algorithm when handling different probabilities of missingness. The missingness imposing is implemented using the R packages SIMSEM[29]. The considered missingness probabilities are 10%, 20%, 30%, 40%, and 50% for the instances. The missingness is imposed into 20%, 40%, and 60% of the features. Here only the results on features with 40% missingness are reported as the other results show similar patterns. This ends up with 150 training/test incomplete data sets for each complete data set (30 synthetic incomplete pairs for each of the five missingness probabilities).

Table 4: The data sets.

Data set	**#Features**	**#Instances**	**#Database**
Yacht	7	308	UCI
Forest	13	517	UCI
ENB2012	8	768	UCI
Concrete	9	1030	UCI
Airfoil	6	1503	UCI
Disorders	7	345	OpenML
Kin8nm	9	8191	OpenML
Pol	49	15000	OpenML
Quake	4	2178	OpenML
Libras	91	360	OpenML

4.2 Benchmark Methods

To investigate the imputation performance of the proposed method, it is compared with some state-of-the-art imputation methods including KNN, CART, RF, GPI, and GP-KNN. KNN is used to impute the missing values by averaging the K most similar instances. CART is used for imputation by employing decision trees to predict the missing values based on the non-missing ones. Another method which is based on decision trees approach is RF. It starts by replacing the missing data with the average of the corresponding complete values and then iteratively improves the missing imputation using proximity. For GPI, for each feature having missing values, the data set is divided into two parts. Part 1 contains instances having missing values in this feature and part 2 is for the remaining instances. The complete data (part 2) is then used to build GP-based imputation models for estimating this feature. Finally, the built imputation models are used to predict the missing values. GP-KNN is a combination of both KNN and GPI. It first retrieves the most similar instances to the incomplete one. After that, these instances are used to build the GP-based regression models.

The imputation methods are implemented using the R package MICE [4] with default settings. The symbolic regression process is carried out using the GP approach. The GP framework provided by distributed evolutionary algorithms in python (DEAP) [10] is used for the implementation of all GP-based methods.

The parameters of GP for imputation and symbolic regression are shown in Table 5. Relative square error (RSE) shown in Equation (13) is used as the fitness function for all GP-based methods except the newly proposed WKNN-GP method where WRSE is used as mentioned above (Equation (9)).

$$RSE = \frac{\sum_{i=1}^{n}(y_i - t_i)^2}{\sum_{i=1}^{n}(t_i - \bar{t})^2} \tag{13}$$

where n is the number of instances, y_i is the i^{th} predicted value, t_i is the i^{th} desired value, and $\bar{t}$ is the average of the desired values t_i, $i = 1, 2, 3..., n$.

Table 5: The used values for GP parameters

Parameter	**Value**
Generations	100
Population size	512
Crossover probability	0.8
Mutation probability	0.2
Elitism	Top 10
Selection Method	Tournament Selection
Tournament size	7
Minimum depth	2
Maximum depth	8
Initialization	Ramped half-and-half
Function set	+, -, *, protected / (%)
Terminal set	features and constants $\in U[-1, 1]$

4.3 Performance Measures

To evaluate the imputation methods, there are three different measures. The imputation accuracy, the symbolic regression performance, and the computational time [37].

The imputation accuracy is a measure that evaluates the accuracy of the prediction of the missing values. This measure requires the presence of the original values as ground truth to be compared with the predicted ones. The imputation performance evaluation method is shown in Fig. 2. The original test data are compared with the imputed test data after applying the imputation

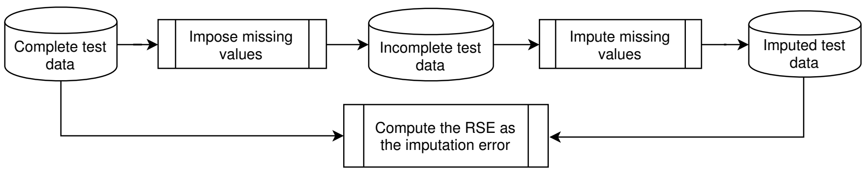

Fig. 2: The imputation performance evaluation.

model. The differences between the predicted values and the ground truth values are computed by the RSE measure (Equation (13)).

The second evaluation measure (symbolic regression performance) could also be done based on testing incomplete data. This scenario simulates the real-life situation when missing values can occur also in the test data when predicting future instances. The adopted evaluation is shown in Fig. 3. It is based on evaluating the symbolic regression performance using the imputed complete test set after applying the imputation method. For each data set, the imputation methods are used to produce complete training and test sets. The training data are then fed into the symbolic regression process resulting in a symbolic regression model. This model is applied to the test data and the obtained regression error is used for comparisons. For the GP-based imputation methods, the imputation models constructed during the training process are used when imputing the test data, whereas the training data set itself is used in the other methods.

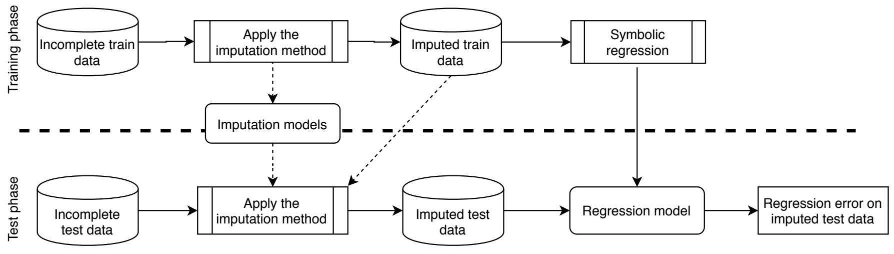

Fig. 3: Symbolic regression performance evaluation.

The goal of the adopted evaluation approach is to simulate the real-world scenario when there are missing values in the test data. This approach has been used in the related work to evaluate the impact of imputation methods when being utilised in real-world machine learning applications [38,14]. In [14], the performance is evaluated on an independent test set generated with the same missingness mechanism as the training data. They then addressed the case of testing on the complete data in the Appendix. In this work we follow this approach as the evaluation is mainly done based on the unseen incomplete test data while the results obtained when testing the models on the original complete data are provided in the supplementary materials.

5 Experimental Results and Analysis

This section shows the experimental results. The results are analyzed and compared with state-of-the-art methods. The comparisons are carried out in terms of the imputation error, the symbolic regression performance, and the computational time.

5.1 Imputation Performance

Following the imputation performance method shown in Fig. 2, the summary of the imputation errors is shown in Fig. 4. Each figure shows the results for one data set where the x-axis refers to the missingness probability (10%, 20%, 30%, 40%, and 50%) and the y-axis is the imputation error measured by the RSE measure (Equation (13)), where the lower RSE error, the better imputation performance. The average of the imputation RSEs of 30 synthetically incomplete data sets with the same missingness probability for each complete data set is shown.

Fig. 4 shows that GP-based methods outperform the other methods in most cases and the proposed WKNN-GP method has the best imputation performance. On nine of the ten data sets, WKNN-GP has the best imputation performance except for ENB2012. On ENB2012, the CART method achieves the smallest imputation error. On eight of the ten data sets, WKNN-GP is significantly better than all the other methods except on the Forest data set, where CART has a similar imputation performance.

The WKNN-GP and GP-KNN methods advance GPI and KNN significantly, which indicates that combining GPI and KNN has a better imputation effort than using them separately. On data sets where KNN has the worst imputation performance (i.e. on the Yacht, Forest, Concrete, Pol, and Libras data sets), WKNN-GP has the best imputation performance. Another indicator of the advantage of the combination is that GP-KNN is ranked second after WKNN-GP with respect to the overall performance.

The performance of the underlying methods KNN and GPI has a high impact on the combined WKNN-GP method. The good performance of GPI along with an acceptable performance of KNN implies a significant improvement of the WKNN-GP performance on the Airfoil data set. However, on the ENB2012 data set, the extremely low performance of KNN decreases the imputation performance of the WKNN-GP method despite the acceptable performance of GPI. Such results may indicate that the similarity between the instances is low, which causes poor performance of KNN and impacts both the GP-KNN and WKNN-GP methods negatively.

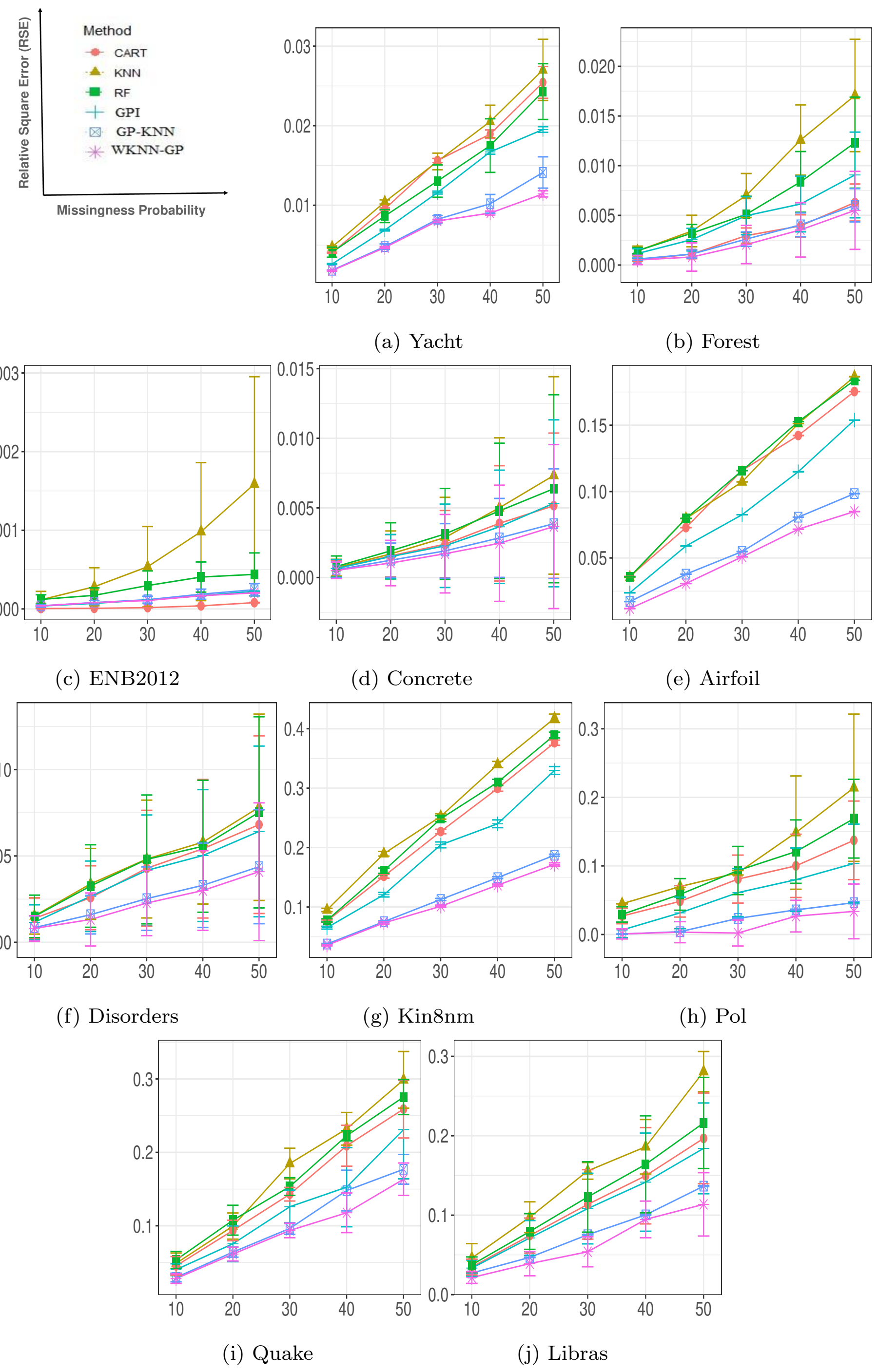

Fig. 4: The imputation errors of using different imputation methods on the synthetic incomplete data sets for different missingness probabilities.

Fig. 4 includes the standard deviation for each imputation method shown as vertical lines. The longer the line is, the higher standard deviation is. The methods with higher standard deviations are less stable than those with lower standard deviations. As shown in Fig. 4, the proposed method shows more stable results than the other methods on most considered data sets. This stability also differs based on the used data set. For example, the Airfoil data set shows stable imputation results regardless of the used imputation method. This might be due to the nature of the data as the features are all numeric in this data set. KNN shows extremely unstable imputation on the ENB2012 and Pol data sets. This might be due to higher dissimilarities between the instances in these data sets which affects KNN as it works by imputation based on the instance-wise similarities.

It can be noticed that the imputation error curves are monotonically increasing. The errors increase with the increase of the missingness ratio. This pattern is common regardless of the imputation method or the used data set. This means that with higher missingness probability, it is less likely to recover the missing values accurately. This is expected as there will be less available data to build the imputation models. Moreover, the differences between the performance of the different methods is affected by the missingness ratio. With less missingness probability, these differences do not seem to be significant.

The significance test results between the proposed method and the benchmark imputation methods are shown in Table 6. The comparisons are carried out using the Wilcoxon non-parametric statistical significance test with a significance level of 0.05. The symbol "+" means WKNN-GP is significantly better, the "-" symbol means WKNN-GP is significantly worse, and "=" indicates no significant difference to the compared method.

From the shown results, it can be noticed that the number of the cases in which the proposed method outperforms each imputation method is higher than the number of getting outperformed cases. Another note is that the differences are less significant for less missingness probabilities. For example the numbers of cases under "=" are relatively large for the 10% missingness probability compared to the corresponding numbers under higher missingness probabilities on most data sets. In other words , the impact of the missingness on the performance of the imputation method is less significant when having less missingness probability.

Table 6: The significance test results of the imputation performance, where the numbers refer to the amount of win(+)/loss(-)/draw(=) cases of WKNN-GP against KNN, CART, RF, GPI and GP-KNN.

Data missing		KNN			CART			RF			GPI			GP-KNN		
		-	=	+	-	=	+	-	=	+	-	=	+	-	=	+
	10	1	22	7	6	13	11	4	5	21	0	14	1	8	17	5
	20	1	17	12	5	18	7	6	12	12	4	16	6	7	13	10
Yacht	30	1	21	8	8	14	8	4	14	12	6	13	3	3	15	12
	40	2	17	11	5	9	16	7	11	12	9	12	5	3	19	8
	50	3	20	7	5	18	7	5	13	12	9	5	9	1	16	13
	10	5	10	15	11	10	9	2	6	22	6	19	5	2	22	6
	20	5	7	18	10	11	9	3	5	22	8	12	10	0	14	16
Forest	30	4	2	24	8	16	6	6	9	15	9	8	13	2	21	7
	40	1	10	19	7	11	12	5	10	15	7	16	7	2	21	7
	50	3	3	24	7	10	13	2	3	25	8	8	14	5	23	2
	10	6	15	3	13	16	1	5	16	9	9	5	16	5	20	5
	20	6	6	9	8	10	12	5	15	10	8	13	9	5	25	0
ENB2012	30	3	2	13	7	11	12	5	21	4	11	14	5	5	24	1
	40	1	1	12	3	13	14	3	22	5	16	12	2	2	22	6
	50	1	3	19	8	18	4	5	13	12	14	8	8	4	23	3
	10	4	12	14	4	15	11	6	13	11	7	14	9	6	17	7
	20	2	10	18	4	12	14	0	12	18	5	12	13	6	14	10
Concrete	30	0	3	27	6	11	13	3	15	12	5	17	8	2	14	14
	40	1	5	24	4	9	17	4	10	16	2	10	18	0	16	14
	50	2	1	27	3	16	11	3	8	19	0	17	13	3	17	10
	10	1	7	22	12	13	5	9	9	12	4	20	6	7	12	11
	20	1	5	24	3	17	10	3	15	12	4	14	12	5	19	6
Airfoil	30	1	3	26	6	21	3	2	13	15	1	13	16	2	19	9
	40	1	0	29	7	11	12	4	15	11	1	16	13	1	12	17
	50	2	1	27	3	15	12	2	15	13	0	20	10	0	10	20
	10	8	14	8	8	13	9	5	10	15	7	15	8	2	19	9
	20	1	11	18	3	14	13	0	3	27	6	11	13	5	20	5
Disorders	30	4	10	16	7	9	14	2	7	21	8	10	12	4	15	11
	40	2	7	21	9	6	15	1	13	16	6	15	9	7	14	9
	50	3	5	22	7	12	11	1	7	22	5	13	12	4	13	13
	10	0	9	21	6	17	7	2	21	7	6	15	9	0	26	4
	20	2	1	27	7	19	4	1	12	17	2	12	16	0	25	5
Kin8nm	30	2	5	23	0	11	19	1	16	13	0	14	16	1	27	2
	40	2	1	27	1	17	12	1	20	9	2	16	12	1	24	5
	50	1	2	27	7	19	4	0	19	11	1	22	7	1	24	5
	10	6	11	13	6	15	9	12	9	9	7	9	14	5	12	13
	20	6	4	20	5	10	15	7	4	19	10	8	12	4	14	12
Pol	30	6	9	15	6	16	8	4	11	15	9	4	17	4	12	14
	40	5	7	18	11	14	5	5	12	13	5	11	14	6	10	14
	50	1	7	22	10	16	4	7	4	19	5	8	17	5	15	10
	10	7	14	9	5	15	10	6	7	17	7	15	8	7	14	9
	20	5	5	20	6	12	12	8	9	13	6	11	13	5	16	9
Libras	30	8	7	15	12	12	6	8	9	13	11	5	14	3	21	6
	40	6	8	16	10	16	4	9	8	13	9	11	10	4	16	10
	50	1	11	18	2	15	13	9	6	15	11	4	15	3	23	4
	10	1	14	15	9	16	5	5	8	17	3	22	5	3	16	11
	20	2	7	21	10	18	2	0	14	16	4	11	15	2	24	4
Quake	30	1	1	28	4	13	13	1	17	12	5	18	7	1	18	11
	40	0	1	29	11	12	7	1	17	12	3	12	15	0	14	16
	50	2	2	26	10	13	7	1	10	19	3	19	8	3	17	10
All	**sum**	**140**	**376**	**934**	**335**	**688**	**477**	**200**	**573**	**727**	**294**	**639**	**529**	**166**	**894**	**440**

5.2 Symbolic Regression Performance

For each data set, the 150 pairs of incomplete training and test data sets are used by each imputation method. The imputed complete pairs are then used for symbolic regression. For each pair, 30 independent symbolic regression experiments are conducted after using each imputation method. The symbolic regression results of using the benchmark imputation methods are then compared with those of using the proposed WKNN-GP method.

The medians of the test RSEs of the symbolic regression models are shown in Table 7. These results show the impact of the imputation methods on the symbolic regression performance with incomplete data. Except for the EBN2012 data set, WKNN-GP leads to better symbolic regression performance than the other methods on all data sets. In general, the GP-based methods provide better symbolic regression results compared with the non-GP ones.

The symbolic regression results from 30 independent runs are used by the Wilcoxon test to compare the WKNN-GP method with the benchmark methods as shown in Table 8. The comparisons show that WKNN-GP achieves better or at least similar symbolic regression performance compared to the other methods in most cases except for the ENB2012 data set. On nine of the ten data sets, the number of having significantly better performance when using WKNN-GP is more than that of having significantly worse results compared to all other methods. The best non-GP method is CART as it outperforms WKNN-GP in more cases than RF and KNN. The worst one is KNN. This is expected as KNN is implemented in a single imputation manner that makes it more sensitive to the outliers when imputing missing values.

The detailed comparisons on each data set considering the missingness probabilities show a marginal significance when having lower missingness probabilities and more significant difference with higher probabilities of missingness. This is because the higher missingness probability, the more likely to get different predictions for the missing values which in turn produces different symbolic regression results.

Table 7: The medians of RSEs of the symbolic regression models learned on imputed training data and tested on the original complete test data, where ratio refers to the missingness probability.

Data	Ratio	KNN	CART	RF	GPI	GP-KNN	WKNN-GP
	10	0.0122	0.011	0.0115	0.0109	0.0101	0.0101
	20	0.0167	0.0143	0.0155	0.0155	0.0151	0.0143
Yacht	30	0.0197	0.0178	0.0199	0.0193	0.0189	0.0178
	40	0.0239	0.0208	0.0238	0.0243	0.0223	0.0208
	50	0.0284	0.0242	0.0288	0.0279	0.0258	0.0242
	10	0.9234	0.9213	0.9219	0.9166	0.9019	0.9007
	20	0.927	0.9262	0.9252	0.9205	0.9054	0.9042
Forest	30	0.9317	0.9298	0.9288	0.9239	0.9099	0.9087
	40	0.9355	0.9341	0.9332	0.9279	0.9129	0.9117
	50	0.9396	0.9387	0.9378	0.9309	0.9168	0.9156
	10	0.092	0.0911	0.0918	0.0914	0.0915	0.0913
	20	0.0968	0.0946	0.0953	0.0959	0.0951	0.095
ENB2012	30	0.1013	0.0989	0.1002	0.1006	0.0999	0.0998
	40	0.1053	0.1028	0.1034	0.1045	0.1042	0.1033
	50	0.1084	0.1072	0.1077	0.1083	0.1084	0.1076
	10	0.4342	0.4323	0.4342	0.4307	0.4305	0.4299
	20	0.4375	0.4361	0.4387	0.4355	0.4334	0.4328
Concrete	30	0.4419	0.4396	0.4433	0.4394	0.4383	0.4377
	40	0.4454	0.4429	0.4475	0.4445	0.4413	0.4407
	50	0.4484	0.4465	0.4506	0.4494	0.4459	0.4453
	10	0.4482	0.4482	0.4486	0.4438	0.4402	0.4396
	20	0.4526	0.4526	0.4534	0.4474	0.4443	0.4437
Airfoil	30	0.4557	0.4572	0.4564	0.4505	0.4483	0.4477
	40	0.4599	0.4603	0.4601	0.4535	0.453	0.4524
	50	0.4634	0.4653	0.4635	0.4585	0.4569	0.4563
	10	0.4674	0.0945	0.0936	0.0931	0.0916	0.0915
	20	0.471	0.0986	0.098	0.0968	0.0964	0.0963
Disorders	30	0.4757	0.1024	0.1018	0.1011	0.1007	0.1006
	40	0.4797	0.1061	0.1067	0.1054	0.104	0.1039
	50	0.4847	0.1107	0.1098	0.1101	0.1083	0.1082
	10	0.7455	0.7461	0.7353	0.7326	0.7239	0.723
	20	0.7501	0.7511	0.7393	0.7358	0.7275	0.7266
Kin8nm	30	0.7534	0.7558	0.7442	0.7408	0.7306	0.7297
	40	0.7569	0.7599	0.748	0.7453	0.7343	0.7333
	50	0.7606	0.7645	0.7513	0.7493	0.7373	0.7363
	10	0.5902	0.5859	0.5868	0.5904	0.5865	0.5873
	20	0.5949	0.5907	0.5905	0.5934	0.5868	0.5851
Pol	30	0.5982	0.5949	0.5938	0.5973	0.5926	0.5918
	40	0.6016	0.5999	0.5981	0.6019	0.5965	0.5957
	50	0.6058	0.6039	0.6016	0.6058	0.6003	0.5995
	10	0.8133	0.7943	0.803	0.7912	0.7791	0.7781
	20	0.8181	0.7983	0.8076	0.7948	0.7828	0.7818
Libras	30	0.8227	0.8016	0.812	0.7998	0.7865	0.7855
	40	0.8276	0.8065	0.8156	0.8036	0.7907	0.7897
	50	0.8318	0.8103	0.8187	0.8072	0.7947	0.7937
	10	0.9148	0.8896	0.9094	0.8782	0.8602	0.8591
	20	0.9181	0.8928	0.9135	0.8832	0.8644	0.8633
Quake	30	0.9226	0.8976	0.9185	0.8866	0.8692	0.8681
	40	0.926	0.9023	0.9235	0.8899	0.873	0.8719
	50	0.9302	0.9064	0.9267	0.8941	0.8767	0.8756

Table 8: The proposed WKNN-GP against the five benchmark imputation methods based on the symbolic regression test performance using imputed data sets.

Data & missing probability (%)		**KNN**			**CART**			**RF**			**GPI**			**GP-KNN**		
		-	=	+	-	=	+	-	=	+	-	=	+	-	=	+
Yacht	10	5	13	12	8	13	9	5	10	15	3	10	17	6	17	7
	20	3	13	14	6	15	9	4	14	12	4	13	13	6	18	6
	30	4	10	16	7	12	11	3	11	16	3	12	15	4	19	7
	40	2	10	18	8	10	12	3	11	16	6	10	14	3	22	5
	50	0	8	22	7	11	12	2	13	15	6	11	13	3	20	7
forest	10	6	13	11	6	17	7	6	12	12	4	11	15	2	24	4
	20	4	11	15	6	18	6	3	12	15	3	10	17	2	20	8
	30	3	8	19	7	14	9	4	15	11	4	8	18	3	22	5
	40	1	8	21	7	15	8	2	13	15	4	8	18	2	26	2
	50	1	5	24	7	14	9	3	8	19	4	7	19	3	25	2
ENB2012	10	7	14	9	11	11	8	7	12	11	4	22	4	3	24	3
	20	4	10	16	9	13	8	2	11	17	3	20	7	3	23	4
	30	3	5	22	12	13	5	2	13	15	3	19	8	4	22	4
	40	0	7	23	15	11	4	2	10	18	4	20	6	4	21	5
	50	0	4	26	16	10	4	2	14	14	4	18	8	4	23	3
Concrete	10	4	16	10	5	19	6	3	13	14	5	12	13	7	18	5
	20	1	14	15	6	15	9	2	12	16	2	11	17	5	17	8
	30	0	9	21	4	16	10	2	13	15	1	14	15	4	19	7
	40	1	7	22	2	16	12	1	9	20	2	13	15	2	18	10
	50	1	6	23	1	20	9	2	12	16	2	13	15	3	15	12
Airfoil	10	0	12	18	4	20	6	5	12	13	7	10	13	7	16	7
	20	0	6	24	4	17	9	2	17	11	2	14	14	4	18	8
	30	0	4	26	3	19	8	3	16	11	3	14	13	3	20	7
	40	0	2	28	3	14	13	3	12	15	2	15	13	0	16	14
	50	0	3	27	2	19	9	3	14	13	2	14	14	1	15	14
Disorders	10	6	16	8	7	15	8	5	11	14	6	14	10	4	20	6
	20	3	15	12	6	13	11	3	10	17	2	8	20	3	21	6
	30	2	13	15	6	13	11	2	11	17	3	9	18	5	19	6
	40	3	8	19	6	16	8	4	8	18	3	11	16	5	18	7
	50	2	8	20	6	15	9	3	13	14	1	10	19	6	15	9
Kin8nm	10	0	15	15	6	18	6	6	15	9	0	19	11	1	26	3
	20	0	4	26	4	16	10	3	14	13	2	17	11	1	29	0
	30	0	0	30	1	19	10	0	11	19	1	21	8	2	27	1
	40	0	0	30	0	18	12	0	12	18	0	18	12	1	25	4
	50	0	0	30	0	22	8	0	21	9	0	17	13	1	24	5
Pol	10	6	12	12	6	13	11	4	10	16	10	11	9	7	14	9
	20	7	9	14	8	11	11	3	9	18	9	10	11	5	16	9
	30	6	9	15	7	9	14	6	11	13	6	12	12	3	18	9
	40	3	9	18	7	11	12	4	13	13	5	12	13	6	16	8
	50	2	7	21	5	12	13	4	12	14	5	10	15	6	15	9
Libras	10	5	14	11	7	15	8	1	12	17	7	8	15	8	16	6
	20	5	11	14	7	13	10	5	11	14	7	10	13	7	18	5
	30	8	11	11	9	10	11	5	14	11	7	12	11	3	20	7
	40	4	8	18	9	9	12	3	11	16	8	10	12	4	20	6
	50	2	9	19	11	6	13	2	12	16	7	12	11	3	21	6
Quake	10	0	17	13	5	20	5	4	11	15	6	13	11	4	21	5
	20	0	8	22	5	17	8	3	17	10	2	17	11	2	24	4
	30	0	4	26	5	19	6	3	15	12	2	16	12	2	24	4
	40	0	2	28	4	14	12	4	11	15	1	17	12	1	20	9
	50	0	0	30	3	19	8	5	11	14	1	14	15	1	22	7
All	**sum**	**114**	**427**	**959**	**306**	**735**	**459**	**158**	**615**	**727**	**188**	**657**	**655**	**179**	**1007**	**314**

Moreover, we also compare the results of WKNN-GP with those obtained by the recent GP methods in the literature on the same data sets. Note that the results from the literature are obtained from complete data sets, which make them different from the ones used in our work. Moreover, the train-test split is also different. The results are shown in Table 9. This table shows results from state-of-the-art symbolic regression method along with the best results obtained when using incomplete data (referred to as "incomplete"). Below is a brief description of the compared methods from the related works.

- In [46], the methods used are GPTIPS: an open-source SR toolbox for MATLAB [35], FFX: fast function extraction [24], EFS: evolutionary feature synthesis [3], and geometric semantic genetic programming with reduced trees (GSGP-Red) [23]. They used a random 0.7/0.3 training/testing ratio in 100 independent runs evaluated by root mean square error (RMSE).
- In [25], four different symbolic regression methods are compared: sequential symbolic regression (SSR), canonical genetic programming (GP), geometric semantic genetic programming (GSGP), and genetic recursive symbolic regression (GRSR). Thee used RMSE in then runs with a 5-fold cross-validations.
- In [42], they showed the results of semantic backpropagation (SB)-based GP without LS (noLS), with LS but independently from SB (iLS), and with LS in synergy with SB (sLS). They split the data set into 0.75/0.25 train/test sets randomly in 30 independent runs evaluated using MSE normalised by variance and multiplied by 100, $NMSE = \frac{MSE*100}{\sigma^2}$.

To convert RMSE to RSE we used the equation $RSE = \frac{RMSE^2}{\sigma^2}$, while $NMSE$ is converted to RSE by $RSE = \frac{NMSE}{100}$. The variance is obtained form [42] and it is calculated from the complete data if not available in the referred study. For the method GPSR, we apply the standard genetic programming-based symbolic regression provided by DEAP [10] on the complete data sets before imposing the synthetic incompleteness.

It is clear that the results on incomplete data are mostly worse than the results of the related work. Such significant difference are expected due to several reasons. The incompleteness adds noise to the data used for learning, which affects the produced models negatively. Another reason is that the methods in the related work represent state-of-the-art algorithms that aim at improving the symbolic regression performance, while in our work we use standard GP-based symbolic regression. However, it can be noticed that the errors are mostly in the same order of magnitude.

Table 9: Symbolic regression results from related studies on complete data sets compared with the results on incomplete data.

Data	Method	RSE	Data	Method	RSE	Data	Method	RSE
Yacht	noLS [42]	0.0094	Airfoil	GPTIPS [25]	0.3601	Concrete	GPTIPS [25]	0.2754
	iLS [42]	0.0062		EFS [25]	0.2759		EFS [25]	0.1482
	sLS [42]	0.0061		FFX [25]	0.2695		FFX [25]	0.1285
	SSR [25]	0.0154		GSGP-Red [25]	3.0988		GSGP-Red [25]	0.2776
	GRSR [25]	0.0075		noLS [42]	0.43		noLS [42]	0.37
	GSGP [25]	0.0268		iLS [42]	0.32		iLS [42]	0.21
	GP	0.0928		sLS [42]	0.29		sLS [42]	0.18
	Incomplete	0.0101		SSR [25]	0.1969		SSR [25]	0.1247
Forest	SSR [25]	0.7414		GRSR [25]	0.3621		GRSR [25]	0.1767
	GRSR [25]	0.7018		GSGP [25]	4.8068		GSGP [25]	0.1937
	GSGP [25]	0.7885		GP	1.9378		GP	0.0858
	GP [25]	0.3735		Incomplete	0.4396		Incomplete	0.4299
	Incomplete	0.9007	Pol	GPSR [10]	0.5714	Disorders	GPSR [10]	0.0871
ENB2012	SSR [25]	0.0651		Incomplete	0.5851		Incomplete	0.0915
	GRSR [25]	0.0854	Libras	GPSR [10]	0.7672	Kin8nm	GPSR [10]	0.7157
	GSGP [25]	0.0357		Incomplete	0.7781		Incomplete	0.723
	GP [25]	0.0885	Quake	GPSR [10]	0.8198			
	Incomplete	0.0913		Incomplete	0.8591			

5.3 Imputation Time

Table 10 shows the average computation time of the WKNN-GP method compared to the other imputation methods when used for imputing a single instance. As shown in Table 10, WKNN-GP is much faster than the other imputation methods except for GPI on all the ten data sets. Most imputation methods are expensive as they require using the training data for estimating missing values in a new instance. However, in our method, the imputation models can be used independently to impute incomplete test instances.

Table 10: The testing computation time (in seconds) of the six imputation methods.

Data Set	**KNN**	**CART**	**RF**	**GPI**	**GP-KNN**	**WKNN-GP**
Yacht	0.12	2.63	5.13	0.0787	467	0.0928
Forest	0.25	10.05	16.89	0.0235	654	0.0397
ENB2012	0.08	5.97	12.56	0.0421	768	0.0546
Concrete	0.09	8.75	17.11	0.0285	985	0.0367
Airfoil	0.18	5.62	12.55	0.0857	1221	0.0889
Disorders	0.07	2.39	5.62	0.0479	478	0.0538
Kin8nm	0.09	87.14	187.08	0.0668	33431	0.07
Pol	0.18	761.19	1103.83	0.0978	3142	0.1135
Quake	0.09	3.89	10.25	0.0437	974	0.0586
Libras	0.16	527.19	630.75	0.0867	11214	0.0984

GPI and WKNN-GP build imputation models in the training process and the built models are used directly to impute new incomplete instances. Thus, the imputation time of the unseen data is decreased dramatically. The slowest one is GP-KNN and it needs to construct an imputation model for each missing value using the training data set. WKNN-GP is more efficient for imputing new incomplete instances than the other imputation methods. On the other hand, WKNN-GP is slightly slower than GPI in many cases. This is a small cost for the much better symbolic regression accuracy and imputation performance.

As seen above, once GP-based models are learned, they can be applied efficiently for imputing values in new instances. This is because GP typically evolves expression trees that are very quick to evaluate compared with other methods (e.g., random forest). However, constructing GP-based imputation models is a time consuming process. The comparison of the training time for the imputation methods are shown in Table 11. This table includes how much time it takes to learn an imputation model with GP against using the other methods. Note that, there is no training time for the KNN method. WKNN-GP spends much loner time than four of the five methods, which is at least 10 times longer than CART. The only method that takes more time than WKNN-GP is GP-KNN. However, the training process of WKNN is affordable in most cases, as it only takes several minutes for model training. In the future, we aim to develop new methods to improve the efficiency of WKNN-GP.

Table 11: The training computation time (in seconds) of the six imputation methods.

Data Set	**KNN**	**CART**	**RF**	**GPI**	**GP-KNN**	**WKNN-GP**
Yacht	0	45.3675	91.0575	153.465	934	625.8656
Forest	0	53.2625	61.6685	73.995	1373.4	395.2929
ENB2012	0	80.595	91.78	96.4834	2150.4	543.6522
Concrete	0	12.46875	31.22575	5.0445	922.1675	366.3394
Airfoil	0	109.59	171.61	131.978	930.4	687.3109
Disorders	0	40.63	120.83	95.8	1099.4	537.4082
Kin8nm	0	206.9575	280.62	120.908	7805.567	697.62
Pol	0	69.648	86.618	91.88	3702.9353	1132.617
Quake	0	62.75	74.0625	69.2	3116.8	583.363
Libras	0	89.223	154.338	157.94	2805	982.2288

5.4 Learned Models

One main advantage of GP-based modelling is the interpret-ability. Although the level of the interpretability depends on some factors such as GP program size, GP produces mathematical models that tend to be interpreteable, which allows more insightful analysis of the obtained results. In Table 12, we show simplified GP imputation models produced by the proposed method for a randomly picked run on the Concrete data set. We can see that some predictive features are used as imputation predictors for different incomplete features frequently. For example, the feature f_2 is used for the three incomplete features. On the other hand, the feature f_7 is used for modelling f_3 but not f_1 and f_6.

Table 12: WKNN-GP imputation models for three incomplete features on the Concrete data set.

Feature	Model (simplified expression)
f_1	$[1 - f_2 f_5 - \frac{f_5^2}{f_2}] * [-f_2^4 f_5 + f_2^2 f_5 - f_2^2 f_5^2 + f_2 f_5^2 - f_2^4 f_5^2 - f_2^3 f_5^2 - f_2^2 f_5^3 + f_2 f_5^3]$
f_3	$\frac{f_5+f_0+1}{f_2+1} + 3f_5 - f_2 + 8$
f_6	$\frac{2f_0 f_2 - 2f_0 + f_2 - 2\frac{f_0 f_2}{f_7} + 5f_5 + f_7 + 1}{2f_2}$

6 Conclusions and Future Work

In this work, a method to improve symbolic regression performance on incomplete data is proposed. This method is called Weighted KNN-GP (WKNN-GP) which integrates two imputation methods: weighted K-nearest neighbour (WKNN) and genetic programming imputation (GPI). Such an integration is presented to utilize both the instance-based similarity of KNN clustering and the feature-based predictability of GP regression.

The evaluation is conducted on real-world data sets considering the imputation accuracy, the symbolic regression performance, and the imputation time. The experimental results show that WKNN-GP outperforms the GPI and KNN methods individually. Moreover, it is significantly better than some state-of-the-art imputation methods. WKNN-GP is efficient for imputing new incomplete instances and it achieves this without sacrificing the imputation accuracy. Although GPI is slightly faster than WKNN-GP, WKNN-GP is significantly better according to both the imputation accuracy and the imputation performance.

For future work, different missingness types (e.g. MCAR and MNAR) will be considered when investigating symbolic regression with missing values. Moreover, the proposed method could be adapted for different machine learning tasks such as classification and clustering.

7 Compliance with Ethical Standards

7.1 Conflict of Interest

The authors declare that they have no conflict of interest.

7.2 Ethical approval

This article does not contain any studies with human participants performed by any of the authors.

8 Declarations

8.1 Funding

Not applicable.

8.2 Conflicts of interest/Competing interests

The authors declare that they have no conflict of interest.

8.3 Availability of data and material

The used data are obtained from the publicly free repositories: UCI and OpenML.

8.4 Code availability

The methods are implemented using python language, mainly based on the open source package DEAP.

References

1. Al-Helali, B., Chen, Q., Xue, B., Zhang, M.: A hybrid GP-KNN imputation for symbolic regression with missing values. In: Australasian Joint Conference on Artificial Intelligence, pp. 345–357. Springer (2018)
2. Anjum, A., Sun, F., Wang, L., Orchard, J.: A novel continuous representation of genetic programmings using recurrent neural networks for symbolic regression. arXiv preprint arXiv:1904.03368 (2019)
3. Arnaldo, I., O'Reilly, U.M., Veeramachaneni, K.: Building predictive models via feature synthesis. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 983–990 (2015)
4. Buuren, S.v., Groothuis-Oudshoorn, K.: mice: Multivariate imputation by chained equations in R. Journal of statistical software pp. 1–68 (2010)
5. Chen, C., Luo, C., Jiang, Z.: Elite bases regression: A real-time algorithm for symbolic regression. In: 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), pp. 529–535. IEEE (2017)
6. Chen, Q.: Improving the generalisation of genetic programming for symbolic regression. Ph.D. thesis, Victoria University of Wellington (2018)
7. Davidson, J.W., Savic, D.A., Walters, G.A.: Symbolic and numerical regression: experiments and applications. Information Sciences **150**(1-2), 95–117 (2003)

8. Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository (2017). URL `http://archive.ics.uci.edu/ml`
9. Donders, A.R.T., Van Der Heijden, G.J., Stijnen, T., Moons, K.G.: A gentle introduction to imputation of missing values. Journal of clinical epidemiology **59**(10), 1087–1091 (2006)
10. Fortin, F.A., Rainville, F.M.D., Gardner, M.A., Parizeau, M., Gagné, C.: Deap: Evolutionary algorithms made easy. Journal of Machine Learning Research **13**(Jul), 2171–2175 (2012)
11. García, J.C.F., Kalenatic, D., Bello, C.A.L.: Missing data imputation in multivariate data by evolutionary algorithms. Computers in Human Behavior **27**(5), 1468–1474 (2011)
12. García-Laencina, P.J., Sancho-Gómez, J.L., Figueiras-Vidal, A.R.: Pattern classification with missing data: a review. Neural Computing and Applications **19**(2), 263–282 (2010)
13. Gautam, C., Ravi, V.: Data imputation via evolutionary computation, clustering and a neural network. Neurocomputing **156**, 134–142 (2015)
14. Ghorbani, A., Zou, J.Y.: Embedding for informative missingness: Deep learning with incomplete data. In: 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 437–445. IEEE (2018)
15. Johnson, C.G.: Artificial immune system programming for symbolic regression. In: European Conference on Genetic Programming, pp. 345–353. Springer (2003)
16. Kammerer, L., Kronberger, G., Burlacu, B., Winkler, S.M., Kommenda, M., Affenzeller, M.: Symbolic regression by exhaustive search: Reducing the search space using syntactical constraints and efficient semantic structure deduplication. In: Genetic Programming Theory and Practice XVII, pp. 79–99. Springer (2020)
17. Koza, J.R.: Genetic Programming II, Automatic Discovery of Reusable Subprograms. MIT Press, Cambridge, MA (1992)
18. Koza, J.R.: Genetic programming as a means for programming computers by natural selection. Statistics and computing **4**(2), 87–112 (1994)
19. Kronberger, G.: Symbolic regression for knowledge discovery: bloat, overfitting, and variable interaction networks. Trauner (2011)
20. Kubalík, J., Žegklitz, J., Derner, E., Babuška, R.: Symbolic regression methods for reinforcement learning. arXiv preprint arXiv:1903.09688 (2019)
21. Lobato, F., Sales, C., Araujo, I., Tadaiesky, V., Dias, L., Ramos, L., Santana, A.: Multi-objective genetic algorithm for missing data imputation. Pattern Recognition Letters **68**, 126–131 (2015)
22. Lobato, F.M., Tadaiesky, V.W., Araújo, I.M., de Santana, Á.L.: An evolutionary missing data imputation method for pattern classification. In: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 1013–1019. ACM (2015)
23. Martins, J.F.B., Oliveira, L.O.V., Miranda, L.F., Casadei, F., Pappa, G.L.: Solving the exponential growth of symbolic regression trees in geometric semantic genetic programming. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1151–1158 (2018)
24. McConaghy, T.: Ffx: Fast, scalable, deterministic symbolic regression technology. In: Genetic Programming Theory and Practice IX, pp. 235–260. Springer (2011)
25. Oliveira, L.O.V., Otero, F.E., Miranda, L.F., Pappa, G.L.: Revisiting the sequential symbolic regression genetic programming. In: 2016 5th Brazilian Conference on Intelligent Systems (BRACIS), pp. 163–168. IEEE (2016)
26. O'Sullivan, J., Ryan, C.: An investigation into the use of different search strategies with grammatical evolution. In: European Conference on Genetic Programming, pp. 268–277. Springer (2002)
27. Patil, D.V., Bichkar, R.: Multiple imputation of missing data with genetic algorithm based techniques. IJCA Special Issue on" Evolutionary Computation for Optimization Techniques pp. 74–78 (2010)
28. Pennachin, C., Looks, M., de Vasconcelos, J.: Improved time series prediction and symbolic regression with affine arithmetic. In: Genetic Programming Theory and Practice IX, pp. 97–112. Springer (2011)

29. Pornprasertmanit, S., Miller, P., Schoemann, A., Quick, C., Jorgensen, T., Pornprasertmanit, M.S.: Package 'simsem' (2016)
30. Priya, R.D., Kuppuswami, S.: A genetic algorithm based approach for imputing missing discrete attribute values in databases. WSEAS Transactions on Information Science and Applications **9**(6), 169–178 (2012)
31. Rubin, D.B.: Inference and missing data. Biometrika **63**(3), 581–592 (1976)
32. Salleh, M.N.M., Samat, N.A.: An imputation for missing data features based on fuzzy swarm approach in heart disease classification. In: International Conference in Swarm Intelligence, pp. 285–292. Springer (2017)
33. Samat, N.A., Salleh, M.N.M.: A study of data imputation using fuzzy c-means with particle swarm optimization. In: International Conference on Soft Computing and Data Mining, pp. 91–100. Springer (2016)
34. Schafer, J.L., Graham, J.W.: Missing data: our view of the state of the art. Psychological methods **7**(2), 147 (2002)
35. Searson, D.P.: Gptips 2: an open-source software platform for symbolic data mining. In: Handbook of genetic programming applications, pp. 551–573. Springer (2015)
36. Takahashi, M., Ito, T.: Multiple imputation of turnover in edinet data: Toward the improvement of imputation for the economic census. Work Session on Statistical Data Editing, UNECE pp. 24–26 (2012)
37. Tran, C.T.: Evolutionary machine learning for classification with incomplete data. Ph.D. thesis, Victoria University of Wellington (2018)
38. Tran, C.T., Zhang, M., Andreae, P.: Multiple imputation for missing data using genetic programming. In: Proceedings of the 2015 annual conference on genetic and evolutionary computation, pp. 583–590. ACM (2015)
39. Tran, C.T., Zhang, M., Andreae, P.: A genetic programming-based imputation method for classification with missing data. In: European Conference on Genetic Programming, pp. 149–163. Springer (2016)
40. Tran, C.T., Zhang, M., Andreae, P., Xue, B.: Multiple imputation and genetic programming for classification with incomplete data. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 521–528. ACM (2017)
41. Vanschoren, J., Van Rijn, J.N., Bischl, B., Torgo, L.: Openml: networked science in machine learning. ACM SIGKDD Explorations Newsletter **15**(2), 49–60 (2014)
42. Virgolin, M., Alderliesten, T., Bosman, P.A.: Linear scaling with and within semantic backpropagation-based genetic programming for symbolic regression. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1084–1092 (2019)
43. Virgolin, M., Alderliesten, T., Witteveen, C., Bosman, P.A.N.: Improving model-based genetic programming for symbolic regression of small expressions. Evolutionary Computation **0**(0), 1–27 (0). DOI 10.1162/evco_a_00278. URL `https://doi.org/10.1162/evco_a_00278`. PMID: 32574084
44. Vladislavleva, E., Smits, G., Den Hertog, D.: On the importance of data balancing for symbolic regression. IEEE Transactions on Evolutionary Computation **14**(2), 252–277 (2010)
45. Wang, Y., Wagner, N., Rondinelli, J.M.: Symbolic regression in materials science. MRS Communications **9**(3), 793–805 (2019)
46. Žegklitz, J., Pošík, P.: Benchmarking state-of-the-art symbolic regression algorithms. Genetic Programming and Evolvable Machines pp. 1–29 (2020)
47. Zelinka, I., Oplatkova, Z., Nolle, L.: Analytic programming–symbolic regression by means of arbitrary evolutionary algorithms. Int. J. of Simulation, Systems, Science and Technology **6**(9), 44–56 (2005)