Genetic Programming with Noise Sensitivity for Imputation Predictor Selection in Symbolic Regression with Incomplete Data

Baligh Al-Helali, Qi Chen, Bing Xue, and Mengjie Zhang
School of Engineering and Computer Science
Victoria University of Wellington, P.O. Box 600, Wellington 6140, New Zealand
{baligh,qi.chen,bing.xue,mengjie.zhang}@ecs.vuw.ac.nz

***Abstract*—This paper presents a feature selection method that incorporates a sensitivity-based single feature importance measure in a context-based feature selection approach. The single-wise importance is based on the sensitivity of the learning performance with respect to adding noise to the predictive features. Genetic programming is used as a context-based selection mechanism, where the selection of features is determined by the change in the performance of the evolved genetic programming models when the feature is injected with noise. Imputation is a key strategy to mitigate the data incompleteness problem. However, it has been rarely investigated for symbolic regression on incomplete data. In this work, an attempt to contribute to filling this gap is presented. The proposed method is applied to selecting imputation predictors (features/variables) in symbolic regression with missing values. The evaluation is performed on real-world data sets considering three performance measures: imputation accuracy, symbolic regression performance, and features' reduction ability. Compared with the benchmark methods, the experimental evaluation shows that the proposed method can achieve an enhanced imputation, improve the symbolic regression performance, and use smaller sets of selected predictors.**

***Index Terms*—Symbolic Regression, Genetic Programming, Incomplete Data, Imputation, Feature Selection**

I. Introduction

Symbolic regression (SR) is the task of finding a symbolic/ mathematical model that best fits a given data set [1]. Compared with traditional methods, it has the advantage of requiring no presumptions on the desired model. Moreover, its "white-box" nature provides desirable interpretability for the learned models. Therefore, symbolic regression has several important applications in various areas such as water resources management [2] and wind energy [3].

Missing values represent a serious problem when learning from real-world data [4]. One widely used approach for dealing with missing values is called imputation. Imputation is the process of predicting the missing values using some estimation models. Imputation methods can be univariate or multivariate. In the univariate methods, the missing values in each feature are imputed using the observed data in the same feature. In contrast, data from different features can be used in multivariate imputation. Consequently, the selection of the features to be involved in the imputation can make a significant difference. The predictive features used for imputing the missing values in an incomplete feature are called imputation predictors.

Feature selection is the process of selecting a subset of relevant/informative features based on predefined criteria [5]. Usually, the criteria imply the impact of the selected subset on the learning performance and the size of this subset. Feature selection has been intensively employed for several machine learning tasks such as classification and clustering. However, a few studies have been published on feature selection for symbolic regression. Moreover, to the best of our knowledge, only two of these studies have considered data incompleteness.

Feature selection methods can be classified as wrapper, filter, and embedded approaches based on the way of involving a learning algorithm in the evaluation procedure [6]. Filter feature selection methods are based on the data properties that can be measured using distance, information, dependency, and consistency [7]. On the other hand, wrapper methods use a learning algorithm for evaluating the features during the manipulation. Unlike filter and wrapper methods, embedded methods build the learning model and select a subset of the features simultaneously.

One way for feature selection is to identify the importance of individual features and select top-ranked features accordingly. Feature importance can be identified by several approaches. For example, it can be measured by the impact of the change in each feature on the learning performance. Such a change can be made by injecting noise into each feature individually [8]. This method is a wrapper-based feature selection method as it requires involving a learning algorithm to measure the performance before and after adding noise.

Genetic programming (GP) is a nature-inspired learning algorithm that simulates the Darwinian evolutionary process to generate computer programs for solving a given problem [1]. It has been successfully used for many tasks. For example, GP is considered as the main approach for symbolic regression. Another interesting aspect of GP is its ability for context-based feature selection. GP is well-known as being a typical example of the embedded feature selection approach. It selects the predictive feature whilst constructing the prediction model. It has been profitably applied for feature selection in different learning tasks such as clustering, classification, and symbolic regression [5]. Moreover, GP has been proposed for dealing with incomplete data achieving encouraging results [9].

One of the main disadvantages of the importance-based feature selection approach is that it does not consider the

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

interaction between different features when constructing prediction models. On the other hand, GP-based feature selection does not consider the individual importance of the involved features. To tackle these disadvantages, a combination of the two approaches is proposed in this work.

In this work, the main goal is to develop a GP-based method for selecting imputation predictors to improve the performance of symbolic regression on incomplete data. In addition to the implicit context-based selection ability of GP, an explicit single-wise importance measure is employed to select the predictors. This measure is based on how sensitive the constructed GP models are to the noise added to the corresponding predictors.

Specific objectives of this work include:

- Developing a predictor selection method that considers both context-wise and single-wise contributions of the predictive features.
- Utilizing the proposed method in imputing the missing values for symbolic regression on incomplete data.
- Evaluating the proposed approach on real-world data sets with comparisons to other approaches using different performance measures.

The rest of this paper is organized as follows. A brief review of the related work is given in Section II. Section III presents the details of the proposed method. In Section IV, the experiment settings are stated and the corresponding experimental results are described in Section V. Finally, this work is concluded in Section VI.

II. Related Work

A. Incomplete Data Imputation

Data incompleteness is a serious problem in regression using real-world data. For example, in the UCI machine learning repository [10], around one-forth the available regression data sets are annotated as having missing values. There are three main types of missing data: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR) [4]. MCAR implies that there is no relationship between the missingness of a value and other data set values, observed or missing. In MAR, the missingness of value is related to some observed data rather than to the missing data itself. However, MNAR means that the missingness is related to the reason it's missing (neither MAR nor MCAR).

Imputation is a key strategy to mitigate the missing data issue [11]. It is used to produce approximate values to fill in the missing data. Imputed data can be produced using two approaches: single imputation and multiple imputation [4]. Single imputation provides a specific value to replace the missing data directly. In contrast, multiple imputation estimates this value from several possible responses based on the variance/confidence interval analysis. After applying imputation, the imputed data can be used for learning by machine learning algorithms in a similar manner that complete data sets are used. Consequently, the imputation performance impacts the whole learning process. Therefore, the adopted imputation approach should be decided carefully.

B. Feature Selection for Symbolic Regression

For high-dimensional symbolic regression, a two-stage feature selection method is presented in [12]. The evolutionary process is split into two phases. The first phase provides a set of candidate important features. This set is then used in the second phase to improve the generalisation of GP models. With the same goal, a different feature selection method is proposed in [13]. This method works by integrating a permutation measure, used in random forest regression, to obtain the importance of features that appear in the GP models. A similar approach is proposed in [14], where the permutation measure is employed for feature selection based on the influence of a feature within a GP model.

In [15], artificial bee colony programming-based feature selection is proposed for symbolic regression. They reported better results than standard GP regarding both the learning ability and generalization performance on synthetic and real-world high-dimensional data sets. In [16], deep learning feature selection is utilized in symbolic regression-classification. It is proposed to enhance GP assisted linear discriminant analysis for multiclass classification problems presented in [17]. However, all these studies consider complete data for evaluating their methods.

C. Feature Selection on Incomplete Data

In [18], an imputation method is proposed based on feature selection and cluster analysis for incomplete high-dimensional data. This method has three steps. First, the dimension is reduced by a clustering-based feature selection algorithm. Second, the selected data are clustered using a parallel k-means method. Finally, the missing values are estimated using data in the same cluster.

In [19], a wrapper-based feature selection method is proposed to improve the performance of a classifier with the ability to classify incomplete data sets. The particle swarm optimisation (PSO) is used for feature selection and the C4.5 method is used for classification. A similar approach is also used for classification with missing values in [20]. In [21], a hybrid method based on fuzzy c-means, mutual information, and regression, is proposed for incomplete data imputation. However, these methods are evaluated on classification tasks.

D. Symbolic Regression with Missing Values

A few studies have been conducted on symbolic regression with missing values. In [22], a GP method for symbolic regression is presented, where missing values are handled through prediction models. However, this method used synthetic data generated by a dynamic model called Lorenz attractor system. An imputation method combining GP and k-nearest neighbor is presented for symbolic regression with missing values in [23]. This method uses GP to construct imputation models for the missing values using instances obtained by KNN. This approach is time consuming especially on large data sets. In [24], a GP-based wrapper imputation method for symbolic regression with incomplete data is proposed. It works by improving the regression prediction of the target variable while constructing GP imputation models for incomplete features. This is done by designing a fitness function that minimizes the target regression error in addition to reducing the imputation errors of estimating the missing values.

Feature selection is employed to select the imputation predictors for symbolic regression with missing values in a few studies. In [25], a GP-based feature selection and ranking method is proposed. It works by constructing GP models for each incomplete feature using other features as predictors. The predictors that appear in the GP models are then ranked based on the fitness values of the models and the occurrence frequency of these predictors. In [26], GP-based method for simultaneous imputation and feature selection is presented. This method constructs the imputation models for the incomplete features and selects their predictive features at the same time. In [27], a complexity measure is employed in GP for model selection and the imputation predictors are selected from the selected models. However, none of these methods considered the impact of individual predictors on the empirical error as a selection criterion. The importance of the predictors is estimated based on the genetic characteristics of the constructed models.

Based on the review above, there is a need for more effort on utilizing feature selection to improving the symbolic regression with missing values.

III. The Proposed Method

The main idea of the proposed method is to integrate a noise sensitivity measure in the GP-based predictor selection process. This section starts by introducing both selection approaches: GP and noise sensitivity, then the proposed method that combines the two approaches is presented.

A. GP-based Predictor Selection

GP performs implicit feature selection as the features used in a GP program represent a set of selected features. For example, in tree-based GP, the target variable is represented as an expression tree in which the leaf nodes can be chosen from a terminal set that contains the input features. Any feature appears in the constructed program is considered as a selected feature by this program.

As GP models can provide mathematical expressions to represent the relationships between input variables and a target variable, it can be used for selecting predictive features for an incomplete feature. Let f be an incomplete feature in a given incomplete data set, D. The data set is reformed to consider this feature as the target variable and the other features as input variables (predictors). Although incomplete features can be considered as predictors, only the complete instances are used. That is, the instances having missing values are ignored, which produces a complete data set, D_f, whose prediction target is f.

After reforming the data set, the standard GP is applied to construct GP prediction models for the feature f. The predictors that appear in these models are considered as the selected imputation predictors for the feature f, $Selected_f$. This process is performed for each incomplete feature getting a set of selected imputation predictor sets for all incomplete features, $Selected_{all}$.

B. Noise Sensitivity-based Predictor Selection

Predictor selection based on sensitivity depends on a straightforward assumption: the prediction target is more sensitive to the change in important predictors than others. The sensitivity can be measured using different techniques. One of these techniques is a noise-based method called feature perturbation. Its idea is that, when perturbed by noise, irrelevant predictors have little influence on the learning performance whereas important predictors impact the performance significantly. As a result, relevant predictors with a high effect on performance under noise are selected.

This method is a wrapper-based feature selection method where a learning algorithm is required for performance evaluation. It uses an iterative process to measure the predictors' sensitivity in three stages. The first stage is to use the wrapper algorithm to build a learning model on a subset of the training data with all available predictors, $D_f^{(subtrain)}$. The noise-free performance is obtained by evaluating the model on a separate holdout data set, $D_f^{(holdout)}$, getting the error $Err_f^{(holdout)}$.

The second stage is to inject noise to a specific predictor, p, in $D_f^{(holdout)}$ getting $D_{f,p}^{(holdout,noisy)}$. After that, the performance of the learned model is evaluated on this noisy holdout data set getting the error $Err_{f,p}^{(holdout,noisy)}$. The two stages are repeated several R times producing two sets of errors. $Errs_{all,f}^{(holdout)}$ is a noise-free error set and $Errs_{all,f,p}^{(holdout,noisy)}$ is the set of errors that measure the impact of adding noise to the predictor p on the prediction of the feature f.

The third stage is to test the significance of the change in the predictor under consideration. This stage is done for each predictor and if the tested change is significant, this predictor is appended to the selected predictor set. The whole process of the three stages is carried out for all incomplete features and the result is a set of selected predictor sets each associated with a specific feature.

C. Noise-Sensitive GP-Based Predictor Selection

The main limitation of noise-based selection is the assumption of predictors' independency. It measures the impact of the change in each predictor independently ignoring the interaction factor between different predictors when used in a prediction model. In contrast, the standard GP selection approach considers the interaction between different predictors rather than the individual importance of each predictor. That is, any predictor appears in the evolved model is selected regardless of its individual importance. Therefore, in this work, a hybrid method between the two approaches is proposed aiming at combining their advantages and suppressing the drawbacks of each approach.

The proposed method has two main steps. The first one is to measure the impact of the noise change in GP selected predictors for a specific incomplete feature, f. This step is shown in Algorithm 1. Secondly, this process is repeated several times and a statistical test is used to assist the significance of the

Algorithm 1: GP-based Predictor Noise Sensitivity Evaluation(D, f)

```
Description: Evaluating predictor sensitivity to noise using GP.
Input      : An incomplete training data set, $D$, and an incomplete feature identifier, $f$.
Output     : Errors before and after adding noise to the predictors of the feature $f$.
Extract a data set of complete instances, $D_f$, considering $f$ as a prediction target variable and the other features, $P_f$, as input variables (predictors);
Split $D_f$ to $D_f^{(subtrain)}$ and $D_f^{(holdout)}$;
$Err_f^{(holdout,noisy)} = \phi$;
Construct GP model, $G_f$, to predict $f$ using the data set $D_f^{(subtrain)}$, i.e. $G_f \approx GP(D_f^{(subtrain)})$;
Evaluate the error of $G_f$ on $D_f^{(holdout)}$ as $Err_f^{(holdout)}$;
foreach Predictor $p \in P_f$ do
    Form $D_{f,p}^{(holdout,noisy)}$ by adding noise into the predictor $p$ in $D_f^{(holdout)}$;
    Evaluate the error of $G_f$ on $D_{f,p}^{(holdout,noisy)}$ as $Err_{f,p}^{(holdout,noisy)}$;
    Append $Err_{f,p}^{(holdout,noisy)}$ to $Errs_f^{(holdout,noisy)}$;
end
return $Err_f^{(holdout)}$, $Errs_f^{(holdout,noisy)}$;
```

Algorithm 2: Selecting predictors for incomplete features using multiple GP runs

```
Input  : An incomplete training data set $D$.
Output : A set of selected predictors for each incomplete feature.
$Selected_{all} = \phi$;
foreach incomplete feature $f$ in $D$ do
    Let $P_f$ be the set of all the available predictors of $f$ ;
    $Selected_f = \phi$;
    Initialize $Errs_{all,f,p}^{(holdout,noisy)} = \phi \; \forall p \in P_f$ ;
    Initialize $Errs_{all,f}^{(holdout)} = \phi$;
    $r = 0$;
    while $r <= R$ do
        Use Algorithm 1 to get the sensitivity errors, $Err_{r,f}^{(holdout)}$ and $Errs_{r,f}^{(holdout,noisy)}$, for the feature $f$ on the data $D$;
        Append $Err_{r,f}^{(holdout)}$ to $Errs_{all,f}^{(holdout)}$;
        foreach Predictor $p \in P_f$ do
            if $p$ in $Errs_{r,f}^{(holdout,noisy)}$ then
                $Err_{f,p}^{(holdout,noisy)}$=$Errs_{r,f,p}^{(holdout,noisy)}$;
            else
                $Err_{f,p}^{(holdout,noisy)}$=$Err_{r,f}^{(holdout)}$;
            end
            Append $Err_{f,p}^{(holdout,noisy)}$ to $Errs_{all,f,p}^{(holdout,noisy)}$;
        end
        $r = r + 1$;
    end
    foreach Predictor $p \in P_f$ do
        $pv$ = significance test($Errs_{all,f}^{(holdout)}$, $Errs_{f,p}^{(holdout,noisy)}$) ;
        if $pv$ is significant then
            Append $p$ to $Selected_f$;
        end
    end
    Append $Selected_f$ to $Selected_{all}$;
end
return $Selected_{all}$;
```

change to decide which predictors deserve to be selected. This step is shown in Algorithm 2.

In this approach, there is a need to decide the noise characteristics. As this approach assumes that the predictors are independent of each other, a Gaussian noise depending on each particular predictor is used as in Eq. (1). However, it is not necessary for the noise to follow a certain distribution since the noise is only used for perturbing the predictors.

$$Noise_f \sim \mathcal{N}(\mu = 0, \sigma^2 = \sigma_f) \tag{1}$$

where, σ_f is the standard deviation of feature f across all training samples.

In Algorithm 1, the training data D is first prepared to consider the incomplete feature f as a prediction target, which produces the data set D_f. After that, D_f is split into two subsets, $D_f^{(subtrain)}$ and $D_f^{(holdout)}$. The data set $D_f^{(subtrain)}$ is used to train a GP model, G_f, to predict the missing values of f, whereas $D_f^{(holdout)}$ is used to evaluate the evolved model getting the prediction error $Err_f^{(holdout)}$.

Next, the sensitivity of the built model, G_f, to noise added to each predictor independently is measured. For each predictive feature, $p \in P_f$, a noisy data set $D_{f,p}^{(holdout,noisy)}$ is generated by injecting noise to the predictor p in $D_f^{(holdout)}$. The impact of this noise is then measured by the error of G_f on $D_{f,p}^{(holdout,noisy)}$ and referred to as $Err_{f,p}^{(holdout,noisy)}$. This process is done for all predictors returning a set that contains all their noise-affected errors, $Errs_f^{(holdout,noisy)}$.

As shown in Algorithm 2, the process of obtaining noise sensitivity errors for each incomplete feature f (Algorithm 1) is repeated R times ($R = 30$ in this work). The output of these runs is a set of R values of the errors obtained from the different runs before adding noise, $Errs_{all,f}^{(holdout)}$, and $|P_f|$ sets of R values of the errors obtained after adding noise to each predictor. Each set of the errors with noise is for a specific predictive feature p and it is denoted as $Errs_{all,f,p}^{(holdout,noisy)}$. Note that, in case the predictor p is not included in the r^{th} GP run, its noise sensitivity error, $Err_{f,p}^{(holdout,noisy)}$, is set to the error before noise $Err_{r,f}^{(holdout)}$, which means zero impact

TABLE I
STATISTICS OF THE DATA SETS

Data set	#Instances	#Features
fri_c0_100_25 (Fri)	100	25
CPMP-2015-runtime-regression (CPMP)	2108	24
Bank32nh (Bank)	8192	33
Selwood	31	54
MIP	1090	145
Mtp	4450	203

TABLE II
GP SETTINGS

Parameter	Value
Generations	50
Population size	1024
Crossover rate	0.9
Mutation rate	0.1
Elitism	Top-5 individual
Selection method	Tournament
Tournament size	7
Maximum depth	9
Initialization	Ramped-half and half
Function set	+, -, *, protected %
Terminal set	predictors and constants $\in U(-1,1)$

of this predictor in this run.

After obtaining the error values before and after adding noise to each predictor for each incomplete feature, a significance test is used to decide whether the predictor has a significant impact on the prediction of f or not. If the difference between $Errs_{all,f}^{(holdout)}$ and $Errs_{f,p}^{(holdout,noisy)}$ is significant then the predictor p is added to the selected predictors of f, $Selected_f$.

IV. Experiment Setup

Six real-world regression data sets obtained from the repository OpenML [28] are used for evaluation in this work. As shown in Table I, these data sets have different numbers of features and instances. Each data set is divided randomly into training and test data sets with a 70:30 ratio. For each data set, 30% Missing At Random (MAR) missingness is imposed on 20% of the features to generate an incomplete copy of the data set. The initial experiments were conducted using the missingness ratios 10%, 30%, 50% for the instances on 10%, 20%, and 30% of the features. However, as similar patterns for the results have been observed, only one combination (30% instance-based ratio on 20% incomplete features). This process is used to generate 30 data set copies for each missingness probability, i.e. 150 incomplete train/test data sets are obtained.

The goodness of the selected predictors is evaluated based on their impact when used by some popular imputation methods. These methods include linear regression (LR), predictive mean matching (PMM), and K-nearest neighbour (KNN) [11]. The synthetic incomplete data sets are generated using the R package SIMSEM [29] and the imputation methods are implemented using the R package Simputation [30] keeping default settings. For GP methods, Table II shows the used settings. The metric used for computing the imputation error and the regression error is relative squared error (RSE) shown in Equation (2).

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

where n is the number of instances, y_i ($\hat{y}_i$) is the target value (predicted value) of the i^{th} instance, and $\bar{y}$ is the average of the target values.

As GP is a stochastic method, the experiments are repeated 30 times for each method then the results are compared statistically. The significance of the difference between the results is measured based on the pair-wise Wilcoxon test with a significance level of 0.05. The proposed method is compared with three methods. The first one is to use all the available features without any feature selection strategy and it is denoted as "Full". This approach is used to evaluate the difference brought when feature selection is employed. Actually, it aims to justify the consideration of feature selection in the first place.

The other methods represent the underlying approaches and they are considered to measure the ability of the proposed method to improve their feature selectability. The second method is to use standard GP for feature selection. The third one is a noise-based feature selection method. As this approach is wrapper-based, following [8], support vector machine (SVM) regressor is used and this method is referred to as noise-based support vector feature selection (NSSV).

V. Results and Discussions

For evaluation, three performance measures are considered: the accuracy of imputing missing values, the symbolic regression performance, and the reduction ratio of selected predictors.

A. Imputation Performance

The imputation performance measures the estimation accuracy of the missing values. It is computed by the RSE error (Eq. (2)) between the original complete data sets and the corresponding imputed data sets. The impact of using predictors selected using the compared selection methods on the imputation results of different imputation methods is shown in Table III. For each data set, the mean of the imputation errors over the corresponding thirty synthetic test incomplete data sets is shown. For the predictors used to impute the incomplete features, column "Full" refers to the use of all the available predictors, "GP" refers to the use of standard GP for predictor selection, "NSSV" means that the predictors are selected by noise-sensitive support vectors, while "NSGP" refers to noise-sensitive GP which implies the use of predictors selected by the proposed method.

The imputation results of different imputation methods when using predictors selected using the considered approaches are shown in Table III. The significance of the difference between the different selection approaches is shown in the column "ST". The symbol "+" ("-") means that the corresponding method outperforms (is outperformed by) the

TABLE III
THE IMPUTATION PERFORMANCE OF DIFFERENT METHODS USING THE COMPARED PREDICTOR SELECTION APPROACHES ON THE TEST DATA SETS

		Full			**GP**			**NSSV**			**NSGP**		
Method	Data	Mean	Sdev	ST	Mean	Sdev	ST	Mean	Sdev	ST	Mean	Sdev	ST
LR	Fri	0.0564	0.0082	(-,-,-)	0.0553	0.0161	(+,-,-)	0.0525	0.0086	(+,+,-)	**0.0519**	0.0093	(+,+,+)
	CPMP	0.1976	0.0115	(-,-,-)	0.185	0.0063	(+,=,-)	0.1843	0.0028	(+,=,-)	**0.1839**	0.0031	(+,+,+)
	Bank	0.1349	0.0155	(=,-,-)	0.1351	0.0031	(=,-,-)	0.1225	0.0092	(+,+,-)	**0.1151**	0.0056	(+,+,+)
	Selwood	0.2141	0.0142	(-,-,-)	0.2106	0.0034	(+,-,-)	**0.2022**	0.0088	(+,+,=)	**0.201**	0.0029	(+,+,=)
	Pah	0.0733	0.0083	(-,-,-)	0.0679	0.0135	(+,-,-)	0.0433	0.0075	(+,+,-)	**0.0391**	0.0072	(+,+,+)
	Mtp	0.1712	0.0158	(-,-,-)	0.1674	0.0116	(+,-,-)	0.1535	0.0023	(+,+,-)	**0.1438**	0.0053	(+,+,+)
KNN	Fri	0.0536	0.0061	(-,-,-)	0.0513	0.0048	(+,-,-)	0.0529	0.0062	(+,+,-)	**0.05**	0.0041	(+,+,+)
	CPMP	0.1764	0.0114	(-,-,-)	0.1501	0.0152	(+,+,-)	0.1519	0.0035	(+,-,-)	**0.1469**	0.0075	(+,+,+)
	Bank	0.1229	0.0126	(-,-,-)	0.1205	0.0025	(+,+,-)	0.1217	0.003	(+,-,-)	**0.1101**	0.0066	(+,+,+)
	Selwood	0.1874	0.0041	(-,-,-)	0.1756	0.0139	(+,-,-)	**0.1532**	0.0051	(+,+,+)	0.1614	0.0025	(+,+,-)
	Pah	0.0545	0.014	(-,-,-)	0.0505	0.0105	(+,-,-)	0.0478	0.0075	(+,+,-)	**0.0374**	0.0104	(+,+,+)
	Mtp	0.1312	0.0175	(=,=,-)	0.1328	0.0094	(=,=,-)	0.1338	0.0022	(=,=,-)	**0.1198**	0.0034	(+,+,+)
PMM	Fri	0.0511	0.0134	(-,-,-)	0.0503	0.0084	(+,=,-)	0.0501	0.0042	(+,=,-)	**0.0481**	0.0107	(+,+,+)
	CPMP	0.1489	0.003	(=,+,-)	0.148	0.003	(=,+,-)	0.1498	0.0021	(-,-,-)	**0.1423**	0.006	(+,+,+)
	Bank	0.1166	0.0102	(-,-,-)	0.1157	0.0156	(+,=,-)	0.1153	0.0068	(+,=,-)	**0.1001**	0.0064	(+,+,+)
	Selwood	0.1551	0.0185	(=,-,-)	0.1554	0.0122	(=,-,-)	0.1523	0.0098	(+,+,-)	**0.1504**	0.0091	(+,+,+)
	Pah	0.0495	0.0073	(-,-,-)	0.0488	0.0124	(+,-,-)	0.0418	0.0084	(+,+,-)	**0.034**	0.0057	(+,+,+)
	Mtp	0.1122	0.0157	(-,-,-)	0.1028	0.0123	(+,+,-)	0.1084	0.0025	(+,-,-)	**0.0981**	0.0079	(+,+,+)

compared method, whereas "=" means no significant difference. These symbols are shown in 3-tuples to show the test sign of the comparison with the other methods in the same order shown in the table.

Table III shows that the use of features selected by NSGP leads to a remarkable improvement in the imputation performance of the three imputation methods. It significantly outperforms the use of the other three methods in almost all considered cases. Although the two underlying methods, GP and NSSV, improve the imputation performance compared to the case of using all features without feature selection in most cases, they fail against each other in several cases. GP significantly outperforms NSSV in 4 cases while NSSV is better in 10 cases. Moreover, they achieve similar performance in 4 cases. The reason of these results can be the variation in the selection mechanism used in each approach. This variation might make some methods more suitable for specific imputation methods on some data sets. GP suits the cases in which the interaction between features is important, while NSSV fits the cases where some features dominate the others. Fundamentally, hybridizing both approaches enables NSGP to work effectively in both situations.

B. Symbolic Regression Performance

On each training-test data set pair imputed using the previous settings, 30 independent symbolic regression experiments are performed. For each imputation method, the test symbolic regression results (computed using RSE) obtained from using different selection strategies are compared. This ends up with 180 comparisons between each pair of selection methods. Table IV shows the number of the cases when comparing the symbolic regression performance associated with different predictor selection methods. Each value in the table refers to the number of times in which the column method significantly outperforms the method in the corresponding row. The "Sum" row is the total number of win cases for each method on each data set, while the "Total" row is the sum of these sums over all data sets.

From Table IV, the superiority of using NSGP regarding the symbolic regression performance can be easily noticed. It has the most win cases against the other methods on almost all data sets. Such results are consistent with the imputation results as methods with better imputation provide better symbolic regression performance. An example of this pattern is that NSSV has the best symbolic regression results when using KNN on the Selwood data set, which is compatible with imputation results as NSSV is the best as well.

Out of 1620 comparisons (30 synthetic incomplete data sets × 6 original data sets × 3 selection methods × 3 imputation methods), NSGP wins 1207 times while it is outperformed in only 110 comparisons. The best results of NSGP are achieved on the Mtp data set with 216 wins (out of 240) and only 12 losses. This might be due to the relatively high number of features and instances in this data set, which makes it more suitable for feature selection. That is, more features means a higher reduction possibility and a higher number of instances provides more information, which in turn increases the ability of the selection methods to provide more stable results. However, the worst results of NSGP are on the Selwood data set with 159 wins. One possible reason is that the Selwood data set has more features than instances, which might limit the learning ability of the proposed method.

On the other hand, the imputation method associated with the best NSGP results is LR, which achieves 410 wins out of 480 comparisons. Such results are due to the regression nature of the LR method. Similar to the GP method, LR relies on predictive features to predict the incomplete ones, which makes it more sensitive to the used predictors in regression. In contrast, when using an instance-based imputation approach, less positive results are observed in NSGP with the KNN imputation method.

C. Predictor Reduction

For the predictor reduction, the average number of selected predictors for all incomplete features using different methods is calculated for each incomplete copy data set. These averages

TABLE IV
NUMBER OF COMPARISONS IN WHICH THE METHOD IN THE COLUMN HAS A SIGNIFICANTLY BETTER SYMBOLIC REGRESSION PERFORMANCE THAN THE METHOD IN THE CORRESPONDING ROW

		LR				**KNN**				**PMM**			
Data	Method	Full	GP	NSSV	NSGP	Full	GP	NSSV	NSGP	Full	GP	NSSV	NSGP
Fri	Full	0	17	21	26	0	18	26	25	0	14	17	26
	GP	1	0	16	23	2	0	23	21	3	0	4	23
	NSSV	1	4	0	17	0	3	0	24	4	4	0	21
	NSGP	0	1	2	0	0	3	2	0	0	2	2	0
	Wins	2	22	39	66	2	24	51	70	7	20	23	70
CPMP	Full	0	18	21	30	0	14	18	26	0	5	2	26
	GP	5	0	7	22	6	0	2	19	4	0	3	19
	NSSV	1	6	0	15	3	13	0	23	16	14	0	25
	NSGP	0	3	3	0	0	5	0	0	1	2	2	0
	Wins	6	27	31	67	9	32	20	68	21	21	7	70
Bank	Full	0	6	20	27	0	17	17	24	0	17	27	25
	GP	4	0	19	23	5	0	5	22	4	0	6	22
	NSSV	2	4	0	21	5	21	0	26	0	4	0	22
	NSGP	1	3	2	0	0	0	1	0	1	5	2	0
	Wins	7	13	41	71	10	38	23	72	5	26	35	69
Selwood	Full	0	13	24	26	0	16	21	21	0	5	19	17
	GP	5	0	23	25	2	0	21	22	8	0	21	20
	NSSV	1	1	0	6	2	3	0	4	3	2	0	18
	NSGP	1	2	4	0	0	1	20	0	4	2	6	0
	Wins	7	16	51	57	4	20	62	47	15	9	46	55
Pah	Full	0	14	25	25	0	17	24	20	0	12	25	26
	GP	6	0	19	24	6	0	19	23	3	0	18	24
	NSSV	0	4	0	25	2	4	0	21	1	4	0	21
	NSGP	1	3	0	0	1	3	2	0	0	4	1	0
	Wins	7	21	44	74	9	24	45	64	4	20	44	71
Mtp	Full	0	11	22	30	0	6	11	24	0	17	20	27
	GP	7	0	24	23	4	0	6	21	3	0	6	25
	NSSV	1	1	0	22	7	9	0	26	4	19	0	18
	NSGP	0	2	3	0	1	2	0	0	0	1	3	0
	Wins	8	14	49	75	12	17	17	71	7	37	29	70
All	Sum	37	113	255	410	46	155	218	392	59	133	184	405

are then averaged over the 30 copies for each original data set and the results are shown in Table V.

TABLE V
THE AVERAGE NUMBER OF SELECTED PREDICTORS BY EACH METHOD ON DIFFERENT DATA SETS

Data set	**Full**	**GP**	**NSSV**	**NSGP**
Fri	25	9	5	5
CPMP	24	7	8	5
Bank	33	16	21	11
Selwood	54	31	16	11
MIP	145	54	67	22
Mtp	203	73	91	21

It is clear that the NSGP method achieves higher predictor reduction ratios than the other methods. This is expected as the idea of NSGP is to select a subset of the sets selected by both GP and NSSV. While NSSV selects all individually important predictors and GP selects all contributor predictors, NSGP selects the important predictors from the contributing predictors. It selects the subset of the predictors selected by GP that includes only the important predictors according to the noise sensitivity measure.

Another way to judge the effectiveness of the feature selection methods is to examine whether the selected features by the method are the actual predictive features or not. However, this measure requires having data sets annotated stating which features are relevant and which ones are not. This condition holds for the data set Fri as only the first five features are used to generate the target variable while the remaining ones are randomly generated features. When evaluated on this data set, both NSSV and NSGP selection methods are able to determine these features successfully.

VI. CONCLUSIONS AND FUTURE WORK

This work presented an improved GP-based imputation predictor selection method with application to symbolic regression on incomplete data. This method combines two feature selection approaches. The first approach considers the contribution of the feature in its selection context, while the other one relays on the importance of individual features regarding a learning process. The former one is represented using GP and the second one is implemented by measuring the sensitivity of the prediction models to noise added to single features.

The proposed method is applied to an important task that has not been adequately investigated. This work is one of the very first studies on predictor selection for imputation in symbolic regression with incomplete data. The experimental work shows that the proposed method takes the advantages of both underlying methods. It selects the most important predictors while taking into account their context. This conclusion can be induced by the gained imputation improvement over these methods. Moreover, it leads to a significant enhancement when utilized for symbolic regression with missing values. Meanwhile, the improvement in both regression and imputation is accompanied by a high reduction in the number of selected predictors.

For future work, this method can be extended to provide a feature ranking method based on the two considered aspects: single-wise importance and context-based contribution. Moreover, this approach can be extended to be applied to different machine learning tasks, e.g. classification. As the main limitation of the proposed method, the time complexity needs to be addressed. One possible option in this direction is to use surrogates for reducing the evaluation time of the evolved models.

References

[1] J. R. Koza, *Genetic Programming II, Automatic Discovery of Reusable Subprograms*. MIT Press, Cambridge, MA, 1992.

[2] P.-S. Ashofteh, O. Bozorg-Haddad, and H. A. Loáiciga, "Logical genetic programming (lgp) application to water resources management," *Environmental Monitoring and Assessment*, vol. 192, no. 1, p. 34, 2020.

[3] P. Valsaraj, D. A. Thumba, K. Asokan, and K. S. Kumar, "Symbolic regression-based improved method for wind speed extrapolation from lower to higher altitudes for wind energy applications," *Applied Energy*, vol. 260, p. 114270, 2020.

[4] A. R. T. Donders, G. J. Van Der Heijden, T. Stijnen, and K. G. Moons, "A gentle introduction to imputation of missing values," *Journal of clinical epidemiology*, vol. 59, no. 10, pp. 1087–1091, 2006.

[5] B. Xue and M. Zhang, "Evolutionary feature manipulation in data mining/big data," *ACM SIGEVOlution*, vol. 10, no. 1, pp. 4–11, 2017.

[6] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on knowledge and data engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[7] M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 3, pp. 131–156, 1997.

[8] L. Chen, D. B. Goldgof, L. O. Hall, and S. A. Eschrich, "Noise-based feature perturbation as a selection method for microarray data," in *International Symposium on Bioinformatics Research and Applications*. Springer, 2007, pp. 237–247.

[9] C. T. Tran, M. Zhang, P. Andreae, B. Xue, and L. T. Bui, "Improving performance of classification on incomplete data using feature selection and clustering," *Applied Soft Computing*, vol. 73, pp. 848–861, 2018.

[10] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[11] K. Heidt, "Comparison of imputation methods for mixed data missing at random," Ph.D. dissertation, 2019.

[12] Q. Chen, B. Xue, B. Niu, and M. Zhang, "Improving generalisation of genetic programming for high-dimensional symbolic regression with feature selection," in *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2016, pp. 3793–3800.

[13] Q. Chen, M. Zhang, and B. Xue, "Feature selection to improve generalization of genetic programming for high-dimensional symbolic regression," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 5, pp. 792–806, 2017.

[14] G. Dick, "Sensitivity-like analysis for feature selection in genetic programming," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2017, pp. 401–408.

[15] S. Arslan and C. Ozturk, "Multi hive artificial bee colony programming for high dimensional symbolic regression with feature selection," *Applied Soft Computing*, vol. 78, pp. 515–527, 2019.

[16] M. F. Korns and T. May, "Strong typing, swarm enhancement, and deep learning feature selection in the pursuit of symbolic regression-classification," in *Genetic Programming Theory and Practice XVI*. Springer, 2019, pp. 59–84.

[17] M. F. Korns, "Evolutionary linear discriminant analysis for multiclass classification problems," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2017, pp. 233–234.

[18] F. Bu, Z. Chen, Q. Zhang, and L. T. Yang, "Incomplete high-dimensional data imputation algorithm using feature selection and clustering analysis on cloud," *The Journal of Supercomputing*, vol. 72, no. 8, pp. 2977–2990, 2016.

[19] C. T. Tran, M. Zhang, P. Andreae, and B. Xue, "Improving performance for classification with incomplete data using wrapper-based feature selection," *Evolutionary Intelligence*, vol. 9, no. 3, pp. 81–94, 2016.

[20] ——, "A wrapper feature selection approach to classification with missing data," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2016, pp. 685–700.

[21] A. M. Sefidian and N. Daneshpour, "Missing value imputation using a novel grey based fuzzy c-means, mutual information based feature selection, and regression model," *Expert Systems with Applications*, vol. 115, pp. 68–94, 2019.

[22] T. Brandejsky, "Model identification from incomplete data set describing state variable subset only–the problem of optimizing and predicting heuristic incorporation into evolutionary system," in *Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems*. Springer, 2013, pp. 181–189.

[23] B. Al-Helali, Q. Chen, B. Xue, and M. Zhang, "A hybrid GP-KNN imputation for symbolic regression with missing values," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2018, pp. 345–357.

[24] ——, "A genetic programming-based wrapper imputation method for symbolic regression with incomplete data," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2019, pp. 2395–2402.

[25] ——, "Genetic programming for imputation predictor selection and ranking in symbolic regression with high-dimensional incomplete data," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2019, pp. 523–535.

[26] ——, "Genetic programming-based simultaneous feature selection and imputation for symbolic regression with incomplete data," in *Asian Conference on Pattern Recognition*. Springer, 2019, pp. 566–579.

[27] ——, "Hessian complexity measure for genetic programming-based imputation predictor selection in symbolic regression with incomplete data," in *EuroGP 2020: Proceedings of the 23rd European Conference on Genetic Programming*, ser. LNCS, T. Hu, N. Lourenco, and E. Medvet, Eds., vol. 12101. Seville, Spain: Springer Verlag, 15-17 Apr. 2020, pp. 1–17.

[28] J. Vanschoren, J. N. Van Rijn, B. Bischl, and L. Torgo, "Openml: networked science in machine learning," *ACM SIGKDD Explorations Newsletter*, vol. 15, no. 2, pp. 49–60, 2014.

[29] S. Pornprasertmanit, P. Miller, A. Schoemann, C. Quick, T. Jorgensen, and M. S. Pornprasertmanit, "Package 'simsem'," 2016.

[30] M. van der Loo, "simputation: Simple imputation," *R package version 0.2*, vol. 2, 2017.