Neural Computing and Applications
https://doi.org/10.1007/s00521-022-06958-3

ORIGINAL ARTICLE

A reinforcement learning-based approach for imputing missing data

Saqib Ejaz Awan[1] · Mohammed Bennamoun[1] · Ferdous Sohel[2] · Frank Sanfilippo[3] · Girish Dwivedi[4,5,6]

Received: 15 May 2021 / Accepted: 14 January 2022

Abstract
Missing data is a major problem in real-world datasets, which hinders the performance of data analytics. Conventional data imputation schemes such as univariate single imputation replace missing values in each column with the same approximated value. These univariate single imputation techniques underestimate the variance of the imputed values. On the other hand, multivariate imputation explores the relationships between different columns of data, to impute the missing values. Reinforcement Learning (RL) is a machine learning paradigm where the agent learns by taking actions and receiving rewards in response, to achieve its goal. In this work, we propose an RL-based approach to impute missing data by learning a policy to impute data through an action-reward-based experience. Our approach imputes missing values in a column by working only on the same column (similar to univariate single imputation) but imputes the missing values in the column with different values thus keeping the variance in the imputed values. We report superior performance of our approach, compared with other imputation techniques, on a number of datasets.

Keywords Missing data · Reinforcement learning · Imputation

1 Introduction

Missing data is a common problem in real-life datasets. Missing data is caused by incomplete/no measurements due to human/system errors, data corruption, and privacy concerns of users filling data for surveys. Missing data hinders the data analysis because most of the analytical approaches cannot straightforwardly work with incomplete data [41]. Usually, the data are pre-processed to overcome this problem. As such, the goal of data pre-processing is to produce a high-quality dataset without missing values. Such preprocessing techniques include imputation, a term used for handling missing values by replacing missing data with substitute values. Given the relevance of missing data in real-life datasets, missing value imputation has received considerable attention and many imputation methods have been proposed in the literature [17, 21].

✉ Saqib Ejaz Awan
saqibejaz.awan@research.uwa.edu.au

Mohammed Bennamoun
mohammed.bennamoun@uwa.edu.au

Ferdous Sohel
f.sohel@murdoch.edu.au

Frank Sanfilippo
frank.sanfilippo@uwa.edu.au

Girish Dwivedi
girish.dwivedi@perkins.uwa.edu.au

[1] Department of Computer Science and Software Engineering, The University of Western Australia, 35 Stirling Highway, Crawley, WA 6009, Australia

[2] Discipline of Information Technology, Murdoch University, 90 South Street, Murdoch, WA 6150, Australia

[3] School of Population and Global Health, The University of Western Australia, 35 Stirling Highway, Crawley, WA 6009, Australia

[4] Harry Perkins Institute of Medical Research, The University of Western Australia, 35 Stirling Highway, Crawley, WA 6009, Australia

[5] Fiona Stanley Hospital, Murdoch, WA 6150, Australia

[6] Medical School, The University of Western Australia, 35 Stirling Highway, Crawley, WA 6009, Australia

Published online: 11 February 2022

Missingness in data can be categorised as [23]: Missing Completely At Random (MCAR), Missing At Random (MAR), and Missing Not At Random (MNAR). Data are classified as MCAR if the missingness of the data occurs entirely at random (with no dependency on other variables). An example of MCAR is a data collected by failed equipment. This type of missingness is not biased towards any factor/variable and hence does not affect the data analysis. Data are MAR when the probability of the missing data depends on the set of observed data values (e.g. the observed values in other columns of tabular data). For instance, older patients might be more likely to forget a data value (hence missing data), than the younger patients. MNAR occurs when the missingness probability depends on the incomplete variables (means the missingness cannot be explained from the observed variables). For example, people with higher income are less likely to reveal it. In this case, an incomes value is missing because it was too high (the reason for a missing column value is associated with the same column). Various strategies are employed to deal with each type of missing data [15]. In this paper, we introduce missingness in the data by randomly removing values across the data, thus our data are MCAR in this work.

A popular approach to deal with missing data is the complete case analysis [13]. This approach considers only those data observations which have no missing values and deletes all the other data. This approach may lead to a substantial loss of important information since in many cases: (1) the data may only be missing from only a few attributes (data from other attributes/columns can be useful, while it is deleted in the complete case analysis), and (2) the data may be missing from a large number of samples (a large amount of (row) data are deleted in this case) contains a large number of missing values. Another commonly used approach, called hot deck imputation, fills the missing values with random values picked from similar non-missing samples [2]. The main drawback of this approach is its lack of ability to preserve the covariance structure in the imputed data [16].

A number of techniques have been applied to solve the missing data imputation problem. There are two main types of imputation techniques: single imputation and multiple imputation [8]. Single imputation approaches estimate the missing values in the data only once, while multiple imputation approaches produce multiple datasets each with an approximation/estimate of the missing values, and the results from all the imputations are consolidated in the final stage to infer the missing values. The single imputation approaches can broadly be categorized as [13]: (1) univariate single imputation approaches such as ad-hoc imputation, nonresponse weighting, and likelihood-based methods; and (2) multivariate single imputation approaches such as k-Nearest Neighbours (kNN), and Random Forests (RF)-based imputation. The univariate imputation approaches replace missing values in a column (of a tabular data) by using the observed values in the same column, whereas the multivariate imputation approaches use the observed values in other columns of the data to estimate the missing values of a column of the data.

Univariate single imputation approaches, in general, impute the missing values in a column of data by using only non-missing values from the same column. The ad hoc imputation aims at maintaining the full data sample by filling the missing values with estimated values. The missing values (in a column of tabular data) are estimated with a single value, in this approach, such as mean or median of the corresponding data feature [20]. The non-response weighting approach also estimates a single value. However, the imputed value, in this case, is a weighted estimate of the population mean or median. The weight is determined by the ratio of the number of samples in each group of data. This approach is suitable only when the data population contains majority samples from one group and a few samples from the other groups. The single imputation approaches fill all the missing values (in a column of tabular data) using only one value, which generally underestimates the errors of data imputation. The likelihood-based methods aim at modelling the missing data mechanism by maximizing the likelihood function of the data [14]. Once the parameters of the likelihood function are estimated, the missing values are produced based on these parameters.

Multivariate single imputation approaches use all the available data across the columns to estimate the missing values. Machine Learning (ML) techniques such as k-NN and RF have been used to address the missing data problem by learning the hidden patterns in the data [19, 33]. Besides the added time complexity of the ML-based approaches, in general, the kNN approach is known to be sensitive to outliers, requires a careful selection of the parameter 'k', and is imprecise in imputing variables which have no dependencies in the dataset [4]. The RF method is known to have biased results at the extreme values of the continuous variables [26].

Multiple imputation replaces the missing values with a set of plausible values by predicting the missing values using the existing data from the other variables [29]. This approach maintains the natural variability and uncertainty in the predicted values. In multiple imputation techniques, the imputation process is iterated several times, each time creating a completed dataset. The completed dataset is then analysed using statistical analysis to generate results. Subsequently, the averaged results are reported. Examples of multiple imputation include techniques based on joint modelling [25], and fully conditional specifications [35]. The former approach assumes a normal distribution of

incomplete variables for imputation, while the latter imputes missing values based on univariate conditional distributions for each incomplete variable given other variables. Despite its sophistication, multiple imputation at times underperforms compared to the simpler (single) imputation approaches [10, 31]. This observation motivated us to focus on single imputation in this work.

Reinforcement Learning (RL) is a type of ML that has robust characteristics to handle the optimization problems by exploring the environment which is formed based on the problem and data. RL enables an agent to learn the best sequence of decisions, through a series of actions and rewards, to achieve an ultimate goal in an environment. We believe that the missing data can be imputed using an RL agent, capable of performing the most suitable action at the right time, to best achieve the goal of approximating the missing data. Therefore, in this work, we propose an RL-based approach to impute missing data, in real-world datasets. Our proposed approach is a univariate single imputation approach. The key aspect of our approach is its ability to estimate missing values without neglecting the variance of the imputed variable, as in the case of conventional univariate single imputation approaches.

2 Related work

A brief categorization of missing data imputation techniques is shown in Fig. 1. Missing data imputation approaches are broadly classified as single imputation approaches, and multiple imputation approaches. A detailed description of these approaches is presented as follows.

2.1 Single imputation approaches

Single imputation approaches estimate each missing value in data with only one value. There are two broad categories of single imputation approaches: univariate single imputation and multivariate single imputation. The detail of these approaches is given as follows.

2.1.1 Univariate single imputation

The most common approach of missing data imputation is the univariate single imputation. Univariate single imputation approaches estimate the missing values, in a column of data, by using the available values from only the same column. Therefore, all the missing values in a column of data are replaced with exactly the same value. Figure 1 shows a number of univariate single imputation approaches, which impute the same value for each missing value in a column (of a tabular data), including the mean, median, most frequent value, and the last observation carried forward imputations [20]. The mean- and median-based imputation approaches impute the missing values in a column with the mean and median of the available values in that column, respectively. The most frequent value-based imputation replaces missing values in a column with the most common value in that column of data. The last observation carried forward imputation replaces missing values with the last observed values. While these approaches are used frequently, they discard the variance of the imputed values, since all the missing values in a column of data are replaced with the same value. These approaches are rigid and are likely to distort the distribution of the imputed variables [18].

2.1.2 Our approach

Our proposed univariate single imputation approach uses a single column during imputation, but estimates dynamic values for each missing value in a column (of tabular data). To the best of our knowledge, this is the first attempt towards a univariate single imputation approach, which replaces missing values in a column while maintaining the variance in the estimated values. The policy learnt by our RL agent guides the imputation process to impute a missing value by using the available values in only the same column of data. A detailed description of our approach is presented in Sect. 3.2.

2.2 Multivariate single imputation

Multivariate single imputation approaches estimate missing values in a column of data, by using the available data in the other columns. These approaches estimate the missing values in a variable by using the relationship among the available data of the other variables. Figure 1 shows a number of multivariate single imputation techniques. One such technique poses missing data imputation task as a matrix completion problem [6]. This technique comprised of a first-order algorithm to fill in the missing entries in low ranked matrices with a minimum nuclear norm. Literature [5] proposed iterative imputation of the missing values of each feature by regressing the values of the remaining features. All these methods are linear in nature, which may not be able to capture the nonlinear relationships between the observed and missing values.

ML techniques such as kNN have also been used to impute missing data [19, 20]. In kNN-based imputation, each missing value is replaced with a value obtained from the related observations of the available dataset. Although this approach is considered an efficient method to fill in the missing data, it tends to distort the true distribution of the data [4].

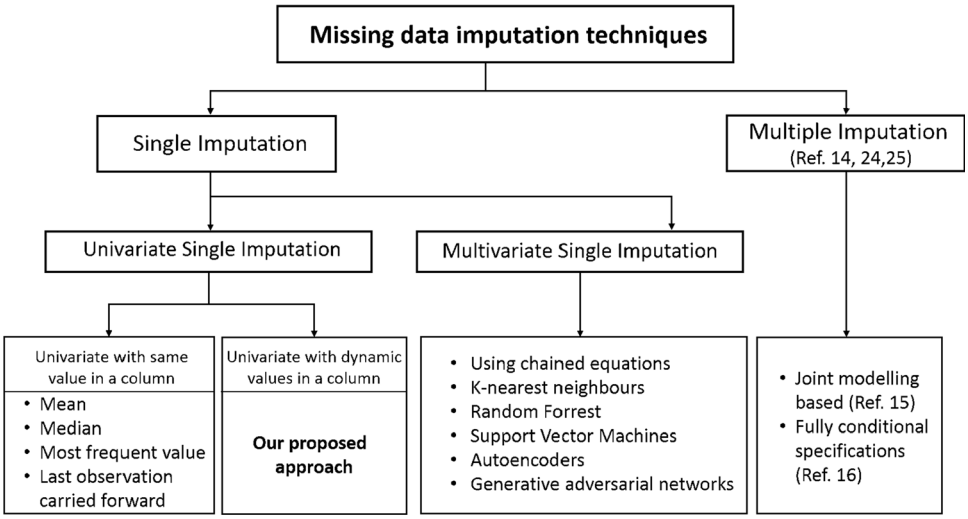

Fig. 1 A broad categorization of missing data imputation techniques

A proximity matrix is also used to impute missing data using RF [33]. In this technique, the data are first imputed using median (for continuous variables) and the most frequently occurring value (for categorical variables). Then, an RF is generated using the filled data and a proximity matrix of size $n * n$ created, where n is the sample size (number of rows in a tabular data). This proximity matrix is then used to impute the originally missing data. The updated data are used to grow another RF and the process is repeated.

Some works have used autoencoders to impute missing data [11, 34]. Gondara et al. proposed a multivariate imputation technique based on deep denoising autoencoders [11]. However, this approach assumed that there is enough complete data to train a model, which might not be the case in real-world datasets. Tran et al. cascaded a series of residual autoencoders to learn the complex relationship from data of different modalities to impute the missing data [34]. This approach combined the strengths of residual learning and autoencoders. Although the autoencoders are empirically effective, these imputation approaches based on autoencoders are heuristic based and it is unclear what mathematical objective is defined for the missing values.

Instead of generating candidate values for the missing data, Smieja et al. presented a general approach to make neural networks process the incomplete data by building a probabilistic model of the incomplete data [28]. Their approach replaced the typical neuron's response in the first hidden layer of a neural network with its expected value to achieve more generalized and accurate activations of the neurons and improve the imputation performance.

A modified Radial Basis Function (RBF) was proposed to generalize the standard Gaussian RBF kernel of Support Vector Machines (SVM) to suite incomplete data [27]. This approach uses the characteristics of the data distribution to model the uncertainty of the missing data to serve for data imputation.

A modified Generative Adversarial Network (GAN) was proposed by Yeh et al. to fill in the missing regions in natural images (known as inpainting) [38]. Their approach was able to learn the representations from the training data and predict the missing patches by using meaningful context. This approach, however, requires complete data in the training phase which is not common in real-world datasets. Since an image is represented with a matrix (or a table) of values (where a value might represent the intensity value of a pixel), it is similar to having a tabular data which does not represent images. Hence, these approaches can also be applied to tabular data.

A Denoising Auto-Encoder-based approach was proposed to impute missing values [24]. Their approach deleted some new missing values in those samples which already have missing values. This extra deletion allowed to better reconstruct the incomplete data by training autoencoders. Their work also introduced a compensation strategy, by adding a balancing parameter in the loss function, to minimize the imbalance in data which was created by the deletion step. Their method achieved similar imputation performance compared with the Multiple Imputation by Chained Equations (MICE), a popular multiple imputation approach.

Popular Generative Adversarial Networks (GANs) [12] have also been used to impute missing data. GANs are a type of machine learning algorithm with generative and discriminator parts, both working in an adversary manner. The generator receives a collection of training examples and learns a probability distribution that generated them. The learnt distribution is then used by the generator to produce new examples. The discriminator part of GANs distinguishes real examples from fake examples, which are generated by the generator. This discrimination is fedback

to the generator to allow it to produce more real-like examples, in an effort to deceive the discriminator. Yoon et al. proposed a GAN-based method, named GAIN, to impute missing data. Their generator completes the missing values given the observed ones, and the discriminator aims to distinguish between true and imputed values [39]. Recently, Awan et al. proposed a class-specific distribution by adapting the popular conditional generative adversarial networks to impute the missing data. Their approach learns class-specific probability distributions in the training phase which allows to impute the missing values more precisely than the GAIN approach [3].

2.3 Multiple imputation approaches

Multiple imputation tries to restore the natural variability in the imputed values. This approach first produces n copies of data [n is typically in the range 5-10 [30]] by imputing missing values in the data n times using a multivariate single imputation approach. Then, each copy of the data is analysed using a standard method (e.g. regressor or a classifier) for complete data. Finally, the results from the analytical method are combined to achieve statistical inference reflecting the uncertainty due to the missing values [22]. MICE is a commonly used approach to generate imputations based on a set of imputation models, one for each variable with missing values [37].

3 The proposed method

The goal of an RL approach is to train an agent, to take decisions at any stage in an environment, to achieve a goal using rewards and punishments. In our work, we aim to train an agent, using RL, to estimate multiple values of missing values in a column of data. Our agent learns to take a series of decisions to make the best estimate of the missing values. A detailed description of RL and our proposed approach is given in the following sections.

3.1 Concept of Reinforcement learning

RL is a machine learning method which is concerned with how an agent should react in an environment. The goal of RL is to train an agent to take a sequence of decisions, using a system of rewards and penalties, to solve a problem by itself. RL achieves its purpose by emulating a scenario and noting the corresponding response of the agent. The agent is rewarded if the response is the desired one and penalized otherwise [32]. Therefore, the next time the agent faces the same situation, it executes a similar action with even more confidence to collect more rewards. Hence, the agent learns "what to do" from good experiences, and "what not to do" from bad experiences.

RL is widely used in robots nowadays, which play a vital role in various applications, such as agriculture, manufacturing, customer service, and health care. Robots in health care provide patients support and assistance in critical situations. These robots are trained by RL, which allows them to learn according to the patients' needs [1].

The core features of a RL paradigm (see Fig. 2) are as follows:

- Observation of the environment: an agent is exposed to the environment.
- Finding yourself in the environment: the situation of the environment that the agent faces, called a state.
- How to act using some strategy: the agent reacts by performing an action to evolve from one state to another.
- Receiving a reward or penalty: After the transition, the agent may receive a reward or penalty in return.
- Learning from experiences: To create a policy, which is the strategy of choosing an action given a state to achieve better outcomes.

3.2 Our proposed approach

Our proposed RL-based approach for missing data imputation is based on the Quality-learning (known as Q-learning approach) [36]. In our RL approach, an agent learns an optimal action-selection policy, from its interaction with the environment, using a Q function [36]. An episode of environment interaction is recorded as (s, a, r, s') using the initial state of the agent (s), the action taken by the agent (a), the reward offered for this action (r), and the resultant state of the agent (s'). Our agent maintains a table $Q[S, A]$ where S is the set of states and A is the set of actions. An experience (s, a, r, s') serves as one data point for the value of $Q[S, A]$. The Q table is updated with each data point using Eq. (1).

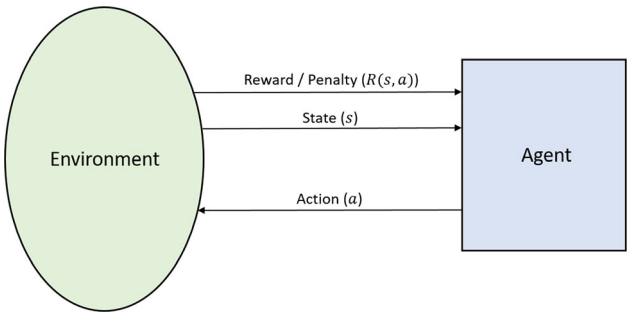

Fig. 2 The components of a typical Reinforcement Learning paradigm

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha[R(s,a) + \gamma \max Q_t(s',a) - Q_t(s,a)] \quad (1)$$

where t represents the current time step, and $t+1$ is the next time step, α is the learning rate $(0 < a \leq 1)$ which determines the amount of update to be made in Q-values in each iteration, γ is the discount factor $(0 \leq \gamma \leq 1)$ which controls the importance given to future rewards. $R(s, a)$ is the current reward for performing an action in the current state. The term $\max Q_t(s', a)$ is the current Q-value estimate of the next best action to be picked. Equation (1) updates the Q-value of the agent's current state and action, by adding the learned value. The learned value is a weighted combination of the reward for taking an action in the current state, and the discounted maximum reward from the next state. This approach motivates the agent to collect maximum rewards and in doing so, learn the best actions to take in a state. The objective of Eq. (1) is to learn a policy to reach the state of lowest error (s_0) from any other state $(s_1 - s_9)$. The Q-values are repeatedly updated using Eq. (1) until a policy is learnt (1000 repetitions in our work).

We initialize the Q-table with zeros initially, which represents the learning of a policy from scratch. Next, an action is chosen from the Q-table and performed using an epsilon greedy strategy. Initially, the values of epsilon are large and the agent explores the environment by choosing actions randomly. The epsilon value gradually decreases and the agent starts to exploit the environment with its experience. In our work, the agent had two actions to choose from: increase the estimated value, or decrease the estimated value.

Algorithm 1: Q-learning (S, A, α, γ)

```
Inputs:
    S = set of states
    A = set of actions
    α = step size
    γ = discount
Output:
    Q[S,A] = Learnt Q-table
Initialize Q[S,A] with zeros
Observe the current state s
Repeat
    Select and perform an action
    Update the imputed value based on the action
    Observe the reward R(s,a) and the new state s'
    Q_{t+1} = Q_t(s,a) + α[R(s,a) + γ max Q_t(s',a) − Q_t(s,a)]
    s ← (s', a)
Untill 1000 iteration
```

In the training phase, the agent knows nothing about the environment initially (i.e. where to look for the best estimate of the missing value). Gradually, the agent learns the manoeuvring and saves it as a policy in the Q-table. Once the Q-table is ready, the agent can start to exploit the environment by taking better actions in each state.

Our proposed RL-based approach for imputing missing data is shown in Fig. 3. The process starts with imputing a singular value for each missing data (for example, the mean value of this column). This approximation determines the state of the imputation, based on the error (how close/far the imputed value is from the ground-truth value). The RL model guides the next imputation value such that the imputed value is pushed towards the state of lower error. Each transition between the states updates the imputation value. At the end of this process, we achieve an imputation value which is very close to the ground truth.

3.2.1 A Markovian formulation of our approach

A Markov Decision Process (MDP) consists of states, actions, rewards, and transitions between the states. In our approach, the set of environment states S is defined as a finite set $\{s_0, s_1, \ldots, s_N\}$, where N is the size of the *state space* S. The size of the state space S is a hyper-parameter, empirically chosen as 10. A state, in our work, is a measure of how far an estimated value is from the actual value. The set of actions A is a finite set $\{a_1, a_2, \ldots, a_K\}$ where K is the size of the *action space*. An action $a \in A$ applicable to a state $s \in S$ is denoted as $A(s)$, where $A(s) \in A$. Each action is used to control the environment's state. In this work, our agent picks one out of two actions, i.e. *increase* the estimated value or *decrease* it. By applying an action $a \in A$ in a state $s \in S$, the environment *transitions* from state s to a new state $s' \in S$. The reward function specifies rewards for being in a state, or doing some action in a state. Our reward function is formally defined as $R : S \times A \times S \rightarrow R$ and represented by the Q-matrix.

An MDP is a sequence of tuples (s, a, r, s'). These sequences of transitions define the *model* of the MDP. A

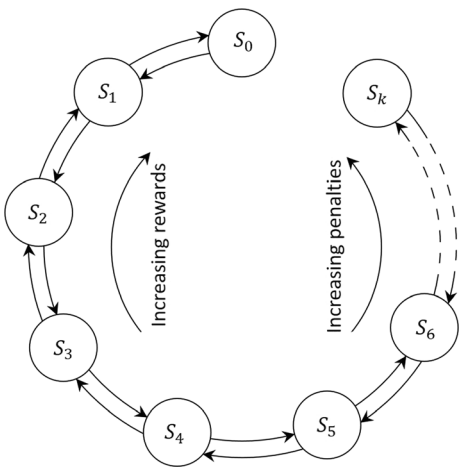

Fig. 3 A Markovian depiction of our RL approach to impute missing values. The training aims at reaching minimum error state, S_0, to minimize the imputation error

pictorial depiction of MDPs is shown in Fig. 3, where the nodes correspond to states and directed edges represent the transitions. Given the MDP, a policy function π outputs for each state $s \in S$ an action $a \in A$. The training begins with a *start state,* e.g. s_0, then the policy π suggests an action a_0, which is performed. A new state s_1 is achieved with this transition and a reward r_0 is collected. This process continues producing $s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, \ldots$, etc., and ends when a goal state, in our case s_0, is achieved. The same process is then repeated with a new *start state*. The learnt policy becomes part of the agent and helps it to control the environment modelled as an MDP.

3.3 A toy example

We present a toy example to demonstrate our proposed approach for imputing missing data. We use the data given in Table 1 as our reference data. The data contains 10 instances of data, each having 4 columns.

We randomly delete 10% of the total data to create missing values (see Table 2). The classical univariate single imputation approaches such as mean, median, and the most frequent value estimate the missing values using statistical measures. A comparison of the statistical-based estimated values and our proposed approach, for the missing values in each column of the toy example data, is given in Table 6 (discussed at the end of this section) (Table 3).

Our RL-based approach starts with learning the policy matrix (Q-matrix) for imputation. For this purpose, we initialize an $n \times n$ matrix of zeros, where n represents the number of states. In this example, we empirically select n to be 10. Each state is based on the error of the imputed value compared with the ground truth value. The rewards matrix R is of the same size as Q. R-matrix contains zero if the path between the corresponding states is viable, and -1 otherwise (path seen in Fig. 3, see Table 4 for R matrix). The error decreases going from state nine (*s9*) towards state zero (*s0*) and vice versa, and our goal is to reach the state with minimum error (*s0*). Therefore, the path of the goal state is set to 100.

Table 1 Toy example: original data

Col 1	Col 2	Col 3	Col 4
0.17	0.26	0.57	1
0.5	0.53	0	0.83
0.83	0	0.57	0.33
0.17	0.39	0.87	0.5
1	0.53	0.14	0.67
0.33	0.84	0.86	0
0.85	0.46	0.17	0.83
0.03	1	0.71	0.5
0.17	0.13	0.86	0.83
0	0.26	1	0.7

Table 2 Toy example: original data with missing values

Col 1	Col 2	Col 3	Col 4
0.17	0.26	0.57	1
0.5	0.53	0	0.83
0.83	0	0.57	?
0.17	?	0.87	0.5
1	0.53	?	0.67
?	0.84	0.86	0
0.85	?	0.17	0.83
0.03	1	0.71	?
0.17	0.13	0.86	0.83
0	0.26	1	0.7

We obtained our trained Q-matrix (shown in Table 4) after 1000 iterations. This matrix contains the policy in the form of a sequence of steps going from a state of higher error to a state of lower error. For each current state (row of the Q-matrix), the column which contains the maximum value is the policy for the next state. Once the Q-matrix is ready, the missing values can be estimated by following the policy given by the Q-matrix and update the estimated value accordingly. The policy is derived from the current state of the agent, followed by the sequence of steps to reach the state zero (*s0*). This process is presented in Table 6 for the missing values in column 2 of our toy example data. The example shows that our proposed approach imputes the two missing values in column 2 of our toy example with two different values. This is a key advantage of our approach since the conventional univariate single imputation techniques lacked variance in the imputed values. A brief description of the process is as follows:

The imputation process starts with an initial estimate of the missing value (column mean in this example). Then, we

Table 3 Initialization of R matrix for Col 2 of our toy example

	s0	s1	s2	s3	s4	s5	s6	s7	s8	s9
s0	100	0	− 1	− 1	− 1	− 1	− 1	− 1	− 1	− 1
s1	100	0	0	− 1	− 1	− 1	− 1	− 1	− 1	− 1
s2	− 1	0	0	0	− 1	− 1	− 1	− 1	− 1	− 1
s3	− 1	− 1	0	0	0	− 1	− 1	− 1	− 1	− 1
s4	− 1	− 1	− 1	0	0	0	− 1	− 1	− 1	− 1
s5	− 1	− 1	− 1	− 1	0	0	0	− 1	− 1	− 1
s6	− 1	− 1	− 1	− 1	− 1	0	0	0	− 1	− 1
s7	− 1	− 1	− 1	− 1	− 1	− 1	0	0	0	− 1
s8	− 1	− 1	− 1	− 1	− 1	− 1	− 1	0	0	0
s9	− 1	− 1	− 1	− 1	− 1	− 1	− 1	− 1	0	0

Table 4 Trained Q-matrix for Col 2 of our toy example

	s0	s1	s2	s3	s4	s5	s6	s7	s8	s9
s0	100	80	0	0	0	0	0	0	0	0
s1	100	80	64	0	0	0	0	0	0	0
s2	0	80	64	51.2	0	0	0	0	0	0
s3	0	0	64	51.2	40.96	0	0	0	0	0
s4	0	0	0	51.2	40.96	32.77	0	0	0	0
s5	0	0	0	0	40.96	32.77	26.21	0	0	0
s6	0	0	0	0	0	32.77	26.21	20.97	0	0
s7	0	0	0	0	0	0	26.21	20.97	16.78	0
s8	0	0	0	0	0	0	0	20.97	16.78	13.42
s9	0	0	0	0	0	0	0	0	16.78	13.42

calculate the error between the estimated value and the ground truth value. The new state of the agent is calculated based on this new imputation error. Then, the Q-matrix is used to get the next move, and the imputation value is updated accordingly. The new value is a weighted update of the current value based on a weight parameter (σ), i.e. $value_{new} = value_{old}(1 + \sigma)$. The sign in this equation is governed by the policy learnt during the training phase. We keep the σ at 0.01 for our toy example. The update in the estimated value is repeated until we reach the state with the minimum error, i.e. *s0*. The estimated value at that point is taken as the imputation value based on our approach.

Table 5 presents a detailed calculation of the imputed values of "Col 2" of the toy example, based on our proposed approach. Table 6 compares the imputation based on mean, median, the most frequent value, kNN, RF, and our proposed approach, on the missing values in each column of the toy example. The first missing value in "Col 2" of the toy example (original value of 0.39) is estimated as 0.44, 0.397, 0.260, 0.463, and 0.330 using mean, median, and the most frequent value, kNN, and RF, respectively. Our proposed RL-based approach imputes this missing value with 0.396. The same mean, median, and the most frequent values are imputed to the second missing value in "Col 2" (original value of 0.460). The kNN and RF-based imputations impute 0.353 and 0.398, while our proposed approach imputes it with 0.452, which is a better approximation of the original value. Table 6 shows that our proposed RL-based approach outperforms the other imputation approaches on the toy example data.

4 Experimental results

4.1 Datasets

We used eight publically available datasets from the UCI Machine Learning Repository [9]. These datasets have been previously used in the literature, e.g. [40]. The details of these datasets are given in Table 7. The *Breast Cancer dataset* contains features, from digitized images, representing characteristics of the cell nuclei such as radius, texture, perimeter, and others. The *Vehicle dataset* is a classification dataset having features extracted from the silhouettes of vehicles. These features include variance, skewness, and kurtosis among others. *Travel dataset* contains features that represent the feedback of customers of the Trip Advisor company. The *Spambase dataset* is a classification dataset whose features come from a collection of emails. The features mostly contain information such as the percentage of occurrence of a specific word in an email, and the length of sequences of consecutive capital letters. *Parkinson dataset* is composed of features representing voice measurements of healthy and Parkinson disease patients. *Letter recognition dataset* contains features from rectangular images representing 26 capital letters in the English alphabet. *Default credit card dataset* is also a classification dataset representing the possibility of default of a customer. The default of a customer is approximated with age, amount of given credit, history of past payments, and other features. *News popularity dataset* contains statistics of online news articles. These statistics include the number of words in the title, number of hyperlinks in the article, the average length of words, and others.

4.2 Performance metrics

The performance metrics, used in this work, to compare our proposed approach for missing data imputation with other available approaches are the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE). These are the most commonly used metrics to estimate the performance of the missing data imputation approaches [7]. MAE is the mean of all the absolute errors between the imputed and ground truth values, as given in Eq. (2).

Table 5 Imputation of missing value using our proposed approach on Col 2 of the toy example data in Tables 1 and 2

Ground truth = 0.390	σ = 0.01		Policy (Table 5)
Estimated = 0.440	Error = 0.050	Current state = s5	Initialize at mean value
Policy = s5→s4			Policy ⇒ negative sign
Estimated = 0.436	Error = 0.046	Current state = s4	= 0.440 (1–0.01)
Policy = s4→s3			= 0.436
Estimated = 0.432	Error = 0.042	Current state = s4	= 0.436 (1–0.01)
Policy = s4→s3			= 0.432
Estimated = 0.428	Error = 0.038	Current state = s3	= 0.432 (1–0.01)
Policy = s3→s2			= 0.428
Estimated = 0.424	Error = 0.034	Current State = s3	= 0.428 (1–0.01)
Policy = s3→s2			= 0.424
Estimated = 0.420	Error = 0.030	Current State = s3	= 0.424 (1–0.01)
Policy = s3→s2			= 0.420
Estimated = 0.416	Error = 0.026	Current State = s2	= 0.420 (1–0.01)
Policy = s2→s1			= 0.416
Estimated = 0.412	Error = 0.022	Current State = s2	= 0.416 (1–0.01)
Policy = s2→s1			= 0.412
Estimated = 0.408	Error = 0.018	Current State = s1	= 0.412 (1–0.01)
Policy = s1→s0			= 0.408
Estimated = 0.404	Error = 0.014	Current State = s1	= 0.408 (1–0.01)
Policy = s1→s0			= 0.404
Estimated = 0.400	Error = 0.010	Current State = s1	= 0.404 (1–0.01)
Policy = s1→s0			= 0.400
Estimated = 0.396	Error = 0.0096	Current State = s0	= 0.400(1–0.01) = 0.396
Ground truth = 0.460	σ = 0.01		
Estimated = 0.440	Error = 0.020	Current State = s2	Initialize at mean value
Policy = s2→s1			Policy ⇒ positive sign
Estimated = 0.444	Error = 0.016	Current State = s1	= 0.440(1 + 0.01) = 0.396
Policy = s1→s0			= 0.444
Estimated = 0.448	Error = 0.012	Current State = s1	= 0.444(1 + 0.01) = 0.39
Policy = s1→s0			=0.448
Estimated = 0.452	Error = 0.008	Current State = s0	=0.448(1 + 0.01) = 0.452

Table 6 Toy example: mean, median, the most frequent value, and our proposed RL-based imputation of missing values from Table 2

	Col 1	Col 2		Col 3	Col 4	
Original value	0.330	0.390	0.460	0.140	0.330	0.500
Mean imputation	0.410	0.440	0.440	0.620	0.670	0.670
Median imputation	0.170	0.397	0.397	0.710	0.765	0.765
Most frequent value imputation	0.170	0.260	0.260	0.570	0.830	0.830
Nearest Neighbour imputation	0.400	0.463	0.353	0.347	0.833	0.453
Random Forest imputation	0.141	0.330	0.398	0.369	0.859	0.338
Our proposed approach	**0.338**	**0.396**	**0.452**	**0.339**	**0.403**	**0.512**

Our approach outperforms (shown as bold) other approaches. The second best results are underlined

Table 7 Characteristics of datasets used in this work

Dataset	No. of instances	No. of attributes
Breast cancer	569	30
Vehicle	946	18
Travel	980	10
Spambase	4,601	57
Parkinson disease	5,875	19
Letter recognition	20,000	17
Default credit card	30,000	24
News popularity	39,644	61

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{n}|x_i - \hat{x_i}| \tag{2}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(x_i - \hat{x_i})^2} \tag{3}$$

where x_i is the ground truth value, $\hat{x_i}$ is the predicted value, and N is the total number of errors. RMSE, given in Eq. (3), represents the square root of the average of the squared differences between the imputed and the ground truth values. While the MAE represents a generic estimate of how far off our imputed values are from the ground truth values, the RMSE is more conscious of the points further away from the mean. This suits us since we want our imputed values to come into the closest-possible vicinity of the ground truth values.

4.3 Experimental setup and results

All the experiments in this work were implemented using Python 3.5, and Scikit-learn 0.22.1. The data were divided into 70% and 30% portions for training and testing, respectively, for each experiment. We created randomly missing data with 5%, 10%, 15%, and 20% proportions across all data in the datasets. All the missing values were replaced with 'nan' during the process. The hyperparameters of Q-learning, such as alpha and discount factor, were selected based on a grid search, in our work. The search spaces of alpha and discount factor were empirically selected as $\{0, 0.001, 0.002, \ldots, 0.5\}$ and $\{0.9, 0.91, 0.92, \ldots, 0.99\}$, respectively. For our approach, the Q-learning was performed over 10,000 iterations to learn the policy for missing data imputation. The Policy matrix (Q-matrix) was initialized with all zeros. Moreover, missing data imputation using our proposed approach was repeated 100 times to check the generalizability of the method. The results were found similar to the performance over a single iteration (presented later in Table 12). For each experiment, we calculated the MAE and RMSE between the imputed and the ground truth values in the test dataset. We used other data imputation techniques such as imputation by mean/median/the most frequent value, nearest neighbour-based imputation, random forest-based imputation, multiple imputation by MICE, GAIN, and CGAIN to compare the performance of our proposed approach for data imputation. The performance of our proposed approach is presented in Tables 8, 9, 10, 11, compared with other imputation methods, for varying amounts of missing data in all the eight UCI datasets used in this work. Our proposed approach has outperformed the other imputation methods on six datasets and remained in the top three for the other two datasets.

As can be seen in Table 8, our RL-based approach performs well compared to other univariate single imputation and ML-based imputation approaches. Our approach produces a MAE of 0.0183 compared to 0.0271, 0.0201, and 0.0208 for mean, median, and the most frequent value-based univariate single imputations, respectively, for Spambase dataset with 5% missing data. For the same settings, the RMSE of our approach is 0.0485 compared to 0.0544, 0.0591, and 0.0615 for mean, median, and the most frequent value-based univariate single imputations. The machine learning-based imputation methods produce a MAE of 0.0309, 0.0309, 0.0286, 0.0501, and 0.0447; and an RMSE of 0.0719, 0.0696, 0.0588, 0.0723, and 0.0611, for kNN-based imputation, RF-based imputation, multiple imputation using chain equations, GAIN, and CGAIN, respectively (see Table 8). The imputation performance of our proposed approach outperforms other approaches with increased proportions of missing data (see Tables 9, 10, 11). The overall imputation performance (measured as MAE and RMSE) decreases for all the methods, as the amount of missing data increases from 5 to 20% (see Tables 8, 9, 10, 11), since less data are available to estimate the missing values. Our proposed approach gives a MAE of 0.0198 compared to 0.0278, 0.0210, 0.0217, 0.0319, 0.0321, 0.0290, 0.0595, and 0.0430, for mean, median, most frequent value-based, kNN-based, RF-based, multiple imputation using chained equations approach, GAIN, and CGAIN, respectively, for Spambase dataset with 20% missing data (see Table 11). The RMSE of our approach, for the same settings, is observed as 0.0527 compared to 0.0593, 0.0635, 0.0667, 0.0750, 0.0715, 0.0620, 0.0764, and 0.0601 for mean, median, most frequent value-based, kNN-based, RF-based, multiple imputation using chained equations approach, GAIN, and CGAIN, respectively. The performance of our proposed approach remained at the top for six datasets, and second-best and third-best for Letter recognition dataset and Breast cancer dataset, respectively.

Table 8 Performance of our proposed approach at 5% missing data based on a single iteration

Dataset	Error type	Single imputation					Multiple imputation (MICE)	GAIN	CGAIN	Proposed univariate approach
		Univariate single imputation			Multivariate single imputation					
		Mean	Median	Most frequent value	Nearest Neighbour	Random Forest				
Spambase	MAE	0.0271	0.0201	0.0208	0.0309	0.0309	0.0286	0.0501	0.0447	**0.0183**
	RMSE	0.0544	0.0591	0.0615	0.0719	0.0696	0.0588	0.0723	0.0611	**0.0485**
Letter recognition	MAE	0.1164	0.1134	0.1166	0.1628	0.1594	0.1406	0.1200	**0.0776**	0.0826
	RMSE	0.1537	0.1545	0.1633	0.2130	0.2092	0.1836	0.1437	**0.1066**	0.1183
Default credit card	MAE	0.0295	0.0263	0.0304	0.0370	0.0370	0.0350	0.2020	0.1971	**0.0212**
	RMSE	0.0552	0.0582	0.0698	0.0735	0.0723	0.0671	0.2428	0.2329	**0.0494**
News popularity	MAE	0.0520	0.0472	0.0873	0.0624	0.0634	0.0610	0.2568	0.1739	**0.0334**
	RMSE	0.1135	0.1166	0.1951	0.1475	0.1533	0.1353	0.2822	0.1964	**0.0866**
Vehicle	MAE	0.1591	0.1553	0.1870	0.1978	0.2041	0.1985	0.1449	0.1362	**0.1181**
	RMSE	0.2080	0.2158	0.2590	0.2573	0.2619	0.2584	0.1881	0.1638	**0.1428**
Parkinson disease	MAE	0.0831	0.0808	0.1023	0.1100	0.1093	0.1037	0.0800	0.0717	**0.0646**
	RMSE	0.1301	0.1315	0.1648	0.1772	0.1752	0.1638	0.1227	0.1186	**0.1074**
Breast cancer	MAE	0.1046	0.1015	0.1393	0.1321	0.1332	0.1371	0.0635	**0.0597**	0.0796
	RMSE	0.1416	0.1466	0.2009	0.1842	0.1856	0.1878	0.0972	**0.0643**	0.1152
Travel	MAE	0.1068	0.1061	0.1201	0.1341	0.1275	0.1267	0.0946	0.0834	**0.0646**
	RMSE	0.1471	0.1498	0.1781	0.1887	0.1806	0.1703	0.1367	0.1255	**0.1128**

MAE = Mean absolute error, RMSE = Root mean squared error. (Best results are in bold, second best results are underlined)

5 Discussions

The univariate single imputation techniques such as imputation with mean, median, or most frequent value do not account for the variations in the imputed values because they impute the same value for each missing value of a column/feature in the dataset. In this work, we have used a reinforcement learning-based approach to account for variations in imputed values and improve the overall estimation of the missing data. Our approach learns a policy, from the training dataset, on how to vary the imputed values to bring them closer to the ground truth value. The learnt policy is used in the testing phase to vary the imputed value to better estimate the missing values. Our approach has worked well compared to the imputation performance of other imputation methods (see Tables 8, 9, 10, 11).

The performance of univariate single imputation techniques deteriorates when the proportion of missing data increases. This is because the singular value (such as mean, median, or the most frequent value) is estimated with fewer data samples and the estimate is likely to be less representative of the entire population. The same trend is observed in other imputation approaches as well as our approach (see Figs. 4 and 5). This trend is reasonable since the ML-based approaches are known to perform well given more data for training. In our approach, we can argue that as the percentage of the missing data increases, the algorithm is not able to learn the best imputation policy which worsens the overall performance.

The performance of our imputation approach is followed by the mean- and median-based imputation techniques, in three datasets (Spambase, Default credit card, and News popularity). This trend is reasonable since the mean and median imputations estimate the missing values with average values. These average values present a reasonable guess given the distribution of the data is normal. As seen in other studies, the mean and median imputation approaches yielded superior results than the multiple imputation approach in our study likely due to the small size of

Table 9 Performance of our proposed approach at 10% missing data based on a single iteration

Dataset	Error type	Single imputation					Multiple imputation (MICE)	GAIN	CGAIN	Proposed univariate approach
		Univariate single imputation			Multivariate single imputation					
		Mean	Median	Most frequent value	Nearest Neighbour	Random Forest				
Spambase	MAE	0.0274	0.0203	0.0211	0.0312	0.0314	0.0288	0.0488	0.0475	**0.0191**
	RMSE	0.0563	0.0606	0.0637	0.0727	0.0701	0.0597	0.0702	0.0664	**0.0494**
Letter recognition	MAE	0.1167	0.1141	0.1177	0.1629	0.1600	0.1426	0.1175	**0.0892**	0.0916
	RMSE	0.1543	0.1552	0.1636	0.2133	0.2099	0.1856	0.1309	**0.1057**	0.1251
Default credit card	MAE	0.0296	0.0264	0.0305	0.0372	0.0377	0.0354	0.1923	0.1804	**0.0223**
	RMSE	0.0557	0.0584	0.0701	0.0727	0.0736	0.0675	0.2109	0.2009	**0.0501**
News popularity	MAE	0.0522	0.0472	0.0874	0.0624	0.0634	0.0616	0.2301	0.1653	**0.0336**
	RMSE	0.1141	0.1172	0.1963	0.1484	0.1536	0.1362	0.2680	0.1937	**0.0889**
Vehicle	MAE	0.1643	0.1586	0.1893	0.1982	0.2059	0.2014	0.1496	0.1407	**0.1226**
	RMSE	0.2148	0.2204	0.2607	0.2653	0.2724	0.2678	0.1900	0.1974	**0.1877**
Parkinson disease	MAE	0.0833	0.0810	0.1025	0.1103	0.1098	0.1056	0.0798	0.0713	**0.0703**
	RMSE	0.1309	0.1328	0.1658	0.1777	0.1756	0.1662	0.1287	0.1213	**0.1148**
Breast cancer	MAE	0.1056	0.1019	0.1421	0.1344	0.1368	0.1394	0.0631	**0.0518**	0.0814
	RMSE	0.1427	0.1469	0.2027	0.1866	0.1886	0.1903	0.0931	**0.0628**	0.1183
Travel	MAE	0.1116	0.1087	0.1229	0.1384	0.1329	0.1283	0.0989	0.0865	**0.0824**
	RMSE	0.1518	0.1521	0.1830	0.1898	0.1809	0.1734	0.1492	0.1316	**0.1373**

MAE = Mean absolute error, RMSE = Root mean squared error. (Best results are in bold, second best results are underlined)

missing data in our data set [10]. The multiple imputation approaches have been shown to produce a more dispersed imputed values thus affecting their performance when used with a small missing data [10].

Multiple imputation, a popular imputation approach from statistics, has not performed well compared to the ML-based approaches. This might be because the multiple imputation approach creates several imputed values for each missing value, where each estimate is regressed from the observed features. The models used to predict an estimate of the missing value, in the case of multiple imputation, cannot exploit the complex relationships among the observed data. This leads to the inadequate performance of this imputation approach.

It should be noted that although GAIN [39] is a supervised approach, the proposed RL approach consistently outperforms GAIN. In addition, compared to a recently proposed CGAIN [3], the proposed RL approach produced superior results on six datasets and slightly inferior results in two datasets. It should further be noted that CGAIN uses a supervised learning approach, which requires a large amount of trained data, while the proposed approach is RL based.

Table 12 shows the performance of our proposed approach compared with other imputation approaches, on different thresholds of missing data, over 100 iterations of data imputation to check the generalizability of our approach. Our proposed approach produces an average MAE (mean $\pm$ standard deviation) of 0.01781 $\pm$ 0.00091, 0.01859 $\pm$ 0.00093, 0.0198 $\pm$ 0.00083, and 0.02017 $\pm$ 0.00054 for 5, 10, 15, and 20% missing data, respectively, in Spambase dataset when the imputation is repeated 100 times. The RMSE, in the same experiment, is recorded as 0.04936 $\pm$ 0.00127, 0.05012 $\pm$ 0.00114, 0.051 $\pm$ 0.0006, and 0.05287 $\pm$ 0.00058 for 5, 10, 15, and 20% missing data, respectively.

Table 13 presents the mean and standard deviation of the original Spambase data, original data with missing values, and data with missing values imputed using our proposed approach. Our proposed RL-based imputation

Table 10 Performance of our proposed approach at 15% missing data based on a single iteration

Dataset	Error type	Single imputation					Multiple imputation (MICE)	GAIN	CGAIN	Proposed univariate approach
		Univariate single imputation			Multivariate single imputation					
		Mean	Median	Most frequent value	Nearest Neighbour	Random Forest				
Spambase	MAE	0.0277	0.0206	0.0213	0.0316	0.0317	0.0289	0.0542	0.0433	**0.0194**
	RMSE	0.0581	0.0621	0.0652	0.0735	0.0707	0.0608	0.0739	0.0607	**0.0511**
Letter recognition	MAE	0.1168	0.1145	0.1182	0.1631	0.1603	0.1440	**0.0851**	0.1113	0.0986
	RMSE	0.1547	0.1559	0.1637	0.2136	0.2107	0.1874	**0.1021**	0.1326	0.1301
Default credit card	MAE	0.0297	0.0265	0.0306	0.0372	0.0373	0.0361	0.2269	0.2137	**0.0233**
	RMSE	0.0562	0.0587	0.0705	0.0739	0.0727	0.0695	0.2442	0.2314	**0.0527**
News popularity	MAE	0.0524	0.0476	0.0876	0.0627	0.0636	0.0617	0.2577	0.1668	**0.0348**
	RMSE	0.1148	0.1180	0.1967	0.1487	0.1536	0.1368	0.2869	0.1992	**0.0961**
Vehicle	MAE	0.1718	0.1646	0.1914	0.2160	0.2205	0.2221	0.1534	0.1461	**0.1362**
	RMSE	0.2206	0.2240	0.2640	0.2926	0.2961	0.3033	0.1917	0.1690	**0.1991**
Parkinson disease	MAE	0.0835	0.0812	0.1026	0.1112	0.1109	0.1059	0.0809	0.0787	**0.0754**
	RMSE	0.1315	0.1334	0.1665	0.1781	0.1760	0.1667	0.01336	0.1286	**0.1221**
Breast cancer	MAE	0.1080	0.1020	0.1438	0.1389	0.1419	0.1408	0.0688	**0.0587**	0.0847
	RMSE	0.1449	0.1486	0.2058	0.1922	0.1942	0.1929	0.0986	**0.0673**	0.1236
Travel	MAE	0.1151	0.1131	0.1232	0.1417	0.1350	0.1324	0.1051	0.0888	**0.0893**
	RMSE	0.1544	0.1556	0.1856	0.1949	0.1867	0.1816	0.1503	0.1470	**0.1424**

MAE = Mean absolute error, RMSE = Root mean squared error. (Best results are in bold, second best results are underlined)

approach maintains the original distribution of the data as mean and standard deviation of 1.662 ± 1.775, 1.668 ± 1.778, and 1.660 ± 1.781 for the original Spambase data, data with missing values, and data with missing values imputed by our approach, respectively, with 5% missing data. With 20% missing data, the values were recorded as 1.662 ± 1.775, 1.673 ± 1.801, and 1.674 ± 1.631 for the original Spambase data, data with missing values, and data with missing values imputed by our approach, respectively. This characteristic of our approach allows to impute accurate values for missing data, which ultimately improves imputation performance. These results show a similar distribution of the data imputed using our approach compared with the original data distribution, at lower rates of missing data (5 and 10%). Understandably, the gap between the original and the imputed data distribution increases when the percentage of missing data increases.

The limitations of our approach include the use of numeric data variables only. Future works will focus on the inclusion of categorical variables in our approach. An extension of this work will focus on the use of additional environment information to guide the agent during the policy learning phase.

6 Conclusion

Missing data imputation has been previously addressed using either a univariate single imputation which discards the variability in the imputed data, or by approximating the missing values using ML models which impute data by exploiting the inherent relationship between the observed features. We proposed an RL approach to learn a good imputation strategy, from experimental trials and the feedback received in response to these trials. Our approach learns the best policy to impute missing data using a trial and reward mechanism for the better approximation of the missing data. The proposed approach has shown superior performance with lower RMSE compared to other data imputation techniques on publically available datasets.

Table 11 Performance of our proposed approach at 20% missing data based on a single iteration

Dataset	Error type	Single imputation					Multiple imputation (MICE)	GAIN	CGAIN	Proposed univariate approach
		Univariate single imputation			Multivariate single imputation					
		Mean	Median	Most frequent value	Nearest Neighbour	Random Forest				
Spambase	MAE	0.0278	0.0210	0.0217	0.0319	0.0321	0.0290	0.0595	0.0430	**0.0198**
	RMSE	0.0593	0.0635	0.0667	0.0750	0.0715	0.0620	0.0764	0.0601	**0.0527**
Letter recognition	MAE	0.1169	0.1148	0.1189	0.1634	0.1612	0.1450	0.1046	**0.0904**	0.1004
	RMSE	0.1553	0.1565	0.1643	0.2141	0.2109	0.1887	0.1302	**0.1040**	0.1373
Default credit card	MAE	0.0298	0.0268	0.0310	0.0378	0.0373	0.0361	0.2255	0.2071	**0.0289**
	RMSE	0.0565	0.0592	0.0710	0.0749	0.0740	0.0697	0.2426	0.2213	**0.0536**
News popularity	MAE	0.0528	0.0478	0.0881	0.0632	0.0641	0.0619	0.2323	0.1631	**0.0381**
	RMSE	0.1156	0.1188	0.1975	0.1494	0.1544	0.1375	0.2686	0.1931	**0.1016**
Vehicle	MAE	0.1735	0.1674	0.2030	0.2274	0.2277	0.2265	0.1641	0.1522	**0.1476**
	RMSE	0.2220	0.2259	0.2756	0.2967	0.2975	0.3052	0.1968	0.1752	**0.2063**
Parkinson disease	MAE	0.0852	0.0829	0.1034	0.1117	0.1114	0.1067	0.0826	0.0811	**0.0797**
	RMSE	0.1339	0.1355	0.1675	0.1794	0.1773	0.1680	0.1330	0.1318	**0.1281**
Breast Cancer	MAE	0.1084	0.1048	0.1442	0.1430	0.1453	0.1468	0.0705	**0.0580**	0.0912
	RMSE	0.1503	0.1549	0.2096	0.1983	0.2011	0.2015	0.1053	**0.0637**	0.1343
Travel	MAE	0.1206	0.1179	0.1280	0.1448	0.1455	0.1387	0.0826	0.0907	**0.0948**
	RMSE	0.1636	0.1652	0.1948	0.2026	0.2015	0.1929	0.1582	0.1501	**0.1483**

MAE = Mean absolute error, RMSE = Root mean squared error. (Best results are in bold, second best results are underlined)

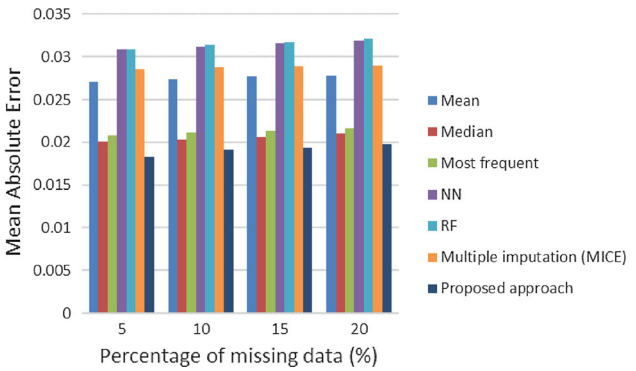

Fig. 4 Performance of different imputation methods on Spambase dataset measured as mean absolute error. NN = Nearest Neighbours, RF = Random Forest, MICE = Multivariate Imputation by Chained Equations

Fig. 5 Performance of different imputation methods on Spambase dataset measured as root mean squared error. NN = Nearest Neighbours, RF = Random Forest, MICE = Multivariate Imputation by Chained Equations

Table 12 Performance of our proposed imputation approach over 100 iterations using different proportions of missing data

Dataset	Error type	Proportion of missing data			
		5%	10%	15%	20%
Spambase	MAE	0.01781 ± 0.00091	0.01859 ± 0.00093	0.0198 ± 0.00083	0.02017 ± 0.00054
	RMSE	0.04936 ± 0.00127	0.05012 ± 0.00114	0.051 ± 0.0006	0.05287 ± 0.00058
Letter recognition	MAE	0.08176 ± 0.00126	0.09167 ± 0.00048	0.09949 ± 0.00131	0.10119 ± 0.00121
	RMSE	0.11744 ± 0.00127	0.12537 ± 0.00068	0.13062 ± 0.00094	0.13815 ± 0.00127
Default credit card	MAE	0.02159 ± 0.0008	0.02288 ± 0.00098	0.02400 ± 0.0012	0.02871 ± 0.00061
	RMSE	0.04957 ± 0.00058	0.04974 ± 0.00078	0.05342 ± 0.00113	0.05356 ± 0.00045
News popularity	MAE	0.03302 ± 0.00079	0.03354 ± 0.00047	0.03498 ± 0.00061	0.03883 ± 0.00115
	RMSE	0.08736 ± 0.00117	0.08860 ± 0.0009	0.09621 ± 0.00053	0.10159 ± 0.00044
Vehicle	MAE	0.11757 ± 0.00095	0.12219 ± 0.00083	0.13576 ± 0.00086	0.14712 ± 0.00089
	RMSE	0.14355 ± 0.00116	0.18733 ± 0.00078	0.19981 ± 0.00113	0.20722 ± 0.00133
Parkinson	MAE	0.06447 ± 0.00055	0.06981 ± 0.00091	0.07497 ± 0.00084	0.07911 ± 0.00101
	RMSE	0.10734 ± 0.00047	0.11484 ± 0.00045	0.12141 ± 0.00111	0.12774 ± 0.00077
Breast cancer	MAE	0.08045 ± 0.00126	0.08171 ± 0.00072	0.08386 ± 0.00125	0.09163 ± 0.00084
	RMSE	0.11486 ± 0.00075	0.11749 ± 0.00123	0.12351 ± 0.00051	0.13337 ± 0.00135
Travel	MAE	0.06534 ± 0.00115	0.08299 ± 0.00102	0.08992 ± 0.00103	0.09561 ± 0.00122
	RMSE	0.11272 ± 0.00049	0.13684 ± 0.00087	0.14184 ± 0.00097	0.14843 ± 0.00056

Table 13 Distribution of data from Spambase dataset: original data vs missing data versus imputed data using our proposed approach

	Percentage of missing data			
Data distribution	5% (mean ± std)	10% (mean ± std)	15% (mean ± std)	20% (mean ± std)
Original	1.662 ± 1.775	1.662 ± 1.775	1.662 ± 1.775	1.662 ± 1.775
Missing	1.668 ± 1.778	1.667 ± 1.778	1.667 ± 1.798	1.673 ± 1.801
Imputed	1.660 ± 1.781	1.662 ± 1.702	1.666 ± 1.681	1.674 ± 1.631

Another advantage of our approach is its power to maintain the original distribution of data during the process, i.e. the distributions of the imputed data and the original data are similar.

Acknowledgements This work is supported by Australian Research Council Grants DP150100294 and DP150104251, and the UWA SIRF scholarship. We thank the contributors of the UCI machine learning repository who collected the data and made it publicly available. We also acknowledge the computing support provided by the NVIDIA Corporation as a Quadro P5000 GPU.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Declaration

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Altameem T, Amoon M, Altameem A (2020) A deep reinforcement learning process based on robotic training to assist mental health patients. Neural Comput Appl 1–10
2. Andridge RR, Little RJ (2010) A review of hot deck imputation for survey non-response. Int Stat Rev 78(1):40–64
3. Awan SE, Bennamoun M, Sohel F, Sanfilippo F, Dwivedi G (2021) Imputation of missing data with class imbalance using conditional generative adversarial networks. Neurocomputing 453:164–171
4. Beretta L, Santaniello A (2016) Nearest neighbor imputation algorithms: a critical evaluation. BMC Med Inf Decis Mak 16(3):74

5. Van Buuren S, Groothuis-Oudshoorn K (2010) MICE: multivariate imputation by chained equations in R. J Stat Softw 45:1–68
6. Cai JF, Candès EJ, Shen Z (2010) A singular value thresholding algorithm for matrix completion. SIAM J Optim 20(4):1956–1982
7. Chai T, Draxler RR (2014) Root mean square error (RMSE) or mean absolute error (MAE)?-arguments against avoiding RMSE in the literature. Geosci Model Dev 7(3):1247–1250
8. Donders ART, Van Der Heijden GJ, Stijnen T, Moons KG (2006) A gentle introduction to imputation of missing values. J Clin Epidemiol 59(10):1087–1091
9. Dua D, Graff C (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml
10. Gómez-Carracedo M, Andrade J, López-Mahía P, Muniategui S, Prada D (2014) A practical comparison of single and multiple imputation methods to handle complex missing data in air quality datasets. Chemom Intell Lab Syst 134:23–33
11. Gondara L, Wang K (2018) MIDA: multiple imputation using denoising autoencoders. In: Pacific-Asia conference on knowledge discovery and data mining (PAKDD 2018). Springer, pp 260–272
12. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. Commun ACM 63(11):139–144
13. He Y (2010) Missing data analysis using multiple imputation: getting to the heart of the matter. Circ Cardiovasc Qual Outcomes 3(1):98–105
14. Hox JJ (1999) A review of current software for handling missing data. Kwant Methoden 20:123–138
15. Kang H (2013) The prevention and handling of the missing data. Korean J Anesthesiol 64(5):402
16. Kim JK, Fuller W (2013) Hot deck imputation for multivariate missing data. In: Proceedings 59th ISI world statistics congress, pp 25–30
17. Lin WC, Tsai CF (2020) Missing value imputation: a review and analysis of the literature (2006–2017). Artif Intell Rev 53(2):1487–1509
18. Lodder P (2013) To impute or not impute: that's the question. Advis Res Methods Sel Top 1–7
19. Mahboob T, Ijaz A, Shahzad A, Kalsoom M (2018) Handling missing values in chronic kidney disease datasets using KNN, K-means and K-medoids algorithms. In: 12th international conference on open source systems and technologies (ICOSST), pp 76–81. IEEE
20. McKnight PE, McKnight KM, Sidani S, Figueredo AJ (2007) Missing data: a gentle introduction, vol 1. Guilford Press
21. Pigott TD (2001) A review of methods for missing data. Educ Res Eval 7(4):353–383
22. Royston P (2004) Multiple imputation of missing values. Stata J 4(3):227–241
23. Rubin DB (1976) Inference and missing data. Biometrika 63(3):581–592
24. Sánchez-Morales A, Sancho-Gómez JL, Martínez-García JA, Figueiras-Vidal AR (2020) Improving deep learning performance with missing values via deletion and compensation. Neural Comput Appl 32(17):13233–13244
25. Schafer JL (1997) Analysis of incomplete multivariate data, vol 1. CRC press
26. Shah AD, Bartlett JW, Carpenter J, Nicholas O, Hemingway H (2014) Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study. Am J Epidemiol 179(6):764–774
27. Śmieja M, Struski Ł, Tabor J, Marzec M (2019) Generalized RBF kernel for incomplete data. Knowl Based Syst 173:150–162
28. Śmieja M, Struski Ł, Tabor J, Zieliński B, Spurek P (2018) Processing of missing data by neural networks. In: Advances in neural information processing systems, pp 2719–2729
29. Sterne JA, White IR, Carlin JB, Spratt M, Royston P, Kenward MG, Wood AM, Carpenter JR (2009) Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. BMJ 338
30. Stuart EA, Azur M, Frangakis C, Leaf P (2009) Multiple imputation with large data sets: a case study of the children's mental health initiative. Am J Epidemiol 169(9):1133–1139
31. Sullivan TR, White IR, Salter AB, Ryan P, Lee KJ (2018) Should multiple imputation be the method of choice for handling missing data in randomized trials? Stat Methods Med Res 27(9):2610–2626
32. Sutton RS, Barto AG (2018) Reinforcement learning: an introduction, vol 2. MIT Press
33. Tang F, Ishwaran H (2017) Random forest missing data algorithms. Stat Anal Data Min ASA Data Sci J 10(6):363–377
34. Tran L, Liu X, Zhou J, Jin R (2017) Missing modalities imputation via cascaded residual autoencoder. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1405–1414
35. Van Buuren S, Brand JP, Groothuis-Oudshoorn CG, Rubin DB (2006) Fully conditional specification in multivariate imputation. J Stat Comput Simul 76(12):1049–1064
36. Watkins CJ, Dayan P (1992) Q-learning. Mach Learn 8(3–4):279–292
37. White IR, Royston P, Wood AM (2011) Multiple imputation using chained equations: issues and guidance for practice. Stat Med 30(4):377–399
38. Yeh IC, Yang KJ, Ting TM (2009) Knowledge discovery on RFM model using Bernoulli sequence. Expert Syst Appl 36(3):5866–5871
39. Yoon J, Jordon J, Schaar M (2018) GAIN: missing data imputation using generative adversarial nets. In: International conference on machine learning, pp 5689–5698. PMLR
40. Zhang H, Xie P, Xing E (2018) Missing value imputation based on deep generative models. arXiv preprint arXiv:1808.01684
41. Zhu B, He C, Liatsis P (2012) A robust missing value imputation method for noisy data. Appl Intell 36(1):61–74