

Article

ChainImputer: A Neural Network-Based Iterative Imputation Method Using Cumulative Features

Wangduk Seo ¹, Timur Khairulov ², Hye-Jin Baek ³ and Jaesung Lee ^{2,*}

¹ Division of AI Computer Science and Engineering, Kyonggi University, Suwon 16227, Republic of Korea; wdseo@kyonggi.ac.kr

² Department of Artificial Intelligence, Chung-Ang University, Seoul 06974, Republic of Korea; timurkhairulov@cau.ac.kr

³ Auton, Anyang 14086, Republic of Korea; iamhyejin@auton.kr

* Correspondence: curseor@cau.ac.kr

Abstract: The goal of missing value imputation is to replace the missing values in the dataset with specific values. In particular, this preprocessing step plays a crucial role in knowledge discovery and data mining, as most data analysis methods assume complete data and cannot be applied directly to datasets with missing entries. Among various approaches, the neural network-based missing value imputation method has recently gained significant attention due to its superior prediction accuracy based on its excellent capability to fit the given training dataset. Specifically, these approaches conventionally begin by applying a naïve missing value imputation to fill all missing entries in the dataset and then train the network on the completed data. Thus, the performance of missing value imputation can be limited because the neural network is trained on an unreliable dataset filled with roughly guessed values. Instead, we may consider an alternative strategy to use only the features without missing values or carefully imputed features obtained during the imputation process, which can be regarded as an asymmetric process because it progressively adds the newly imputed features into the training dataset. In this study, we propose an effective neural network-based imputation method that incrementally constructs a cumulative feature set during training. The experimental results on 25 publicly available datasets showed that the proposed method outperforms conventional methods significantly.

Keywords: missing value imputation; chain imputation; neural network



Academic Editor: Jie Yang

Received: 10 April 2025

Revised: 18 May 2025

Accepted: 26 May 2025

Published: 3 June 2025

Citation: Seo, W.; Khairulov, T.; Baek, H.-J.; Lee, J. ChainImputer: A Neural Network-Based Iterative Imputation Method Using Cumulative Features. *Symmetry* **2025**, *17*, 869. <https://doi.org/10.3390/sym17060869>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Missing values frequently occur in data analysis due to various factors such as incomplete manual data entry, measurement errors, or sensor malfunctions [1]. These unobserved values cannot be directly used in analyzing the characteristics of the data, potentially degrading model performance or even preventing the application of subsequent procedures [2,3]. As a result, most studies considered missing value imputation (MVI) before conducting their analysis, indicating that an accurate imputation is essential for reliable analysis and modeling [4]. Although a variety of naïve MVI methods, such as mean imputation [5], have been widely applied, these methods refer to simple approaches that fill in missing values based solely on the statistical properties of observed data and often fail to capture complex feature interactions [6]. In contrast, neural network (NN)-based approaches have recently attracted significant attention due to their ability to effectively infer missing values from incompletely observed inputs, resulting in high imputation accuracy [7].

The strategy to impute the missing values can be divided into chain and non-chain approaches. Among them, the chain approach can be regarded as an asymmetric process because the chain approach iteratively imputes feature values and progressively adds the newly imputed features into the training dataset. As a result, the input feature space grows step by step—similar to a natural number sequence (1, 2, 3, 4, 5, ...), which introduces an asymmetric structure. In contrast, non-chain approaches can be regarded as symmetric approaches due to maintaining a fixed number of features throughout the process, resulting in a geometrically symmetric training dataset. For example, if we plot the number of features used in each iteration, the chain-based approach forms a right triangle as the feature set grows progressively over time. In this regard, non-chain imputers maintain a constant number of features across iterations, resulting in a rectangular shape. The advantage of the asymmetric chain approach is that it is more focused and detailed since each iteration only tries to impute one feature while using other already imputed features. This can lead to more accurate results at each step. In contrast, the symmetric non-chain approach looks at the whole dataset at once, so it might struggle to predict accurate values because it includes noisy or irrelevant features in the input.

In conventional NN-based MVI methods, the missing values are replaced with the outputs of the NN [8]. To train the NN for missing value prediction, all the features except the target feature are used as a training dataset. Because the training dataset might also contain missing values, a naïve MVI method, such as mean imputation, is typically applied before the training process since conventional NN-based methods cannot handle missing values directly. However, such an approach can lead to inaccurate predictions, as the model is trained on a dataset containing roughly estimated values. This issue becomes more severe when the original dataset contains a large number of missing values, as the NN fits into a significantly distorted dataset.

Ideally, all the distorted features should be excluded from the training dataset to avoid an inaccurate NN model, which can be achieved by considering only features without missing values. However, in practice, there is no guarantee that there will always be a sufficient number of features without missing values, especially when the original dataset contains a large amount of missing data. As an alternative, feature imputation can be performed as an iterative process, where features carefully imputed by the neural network in previous iterations are included in the training dataset to impute the values of the next feature. This approach helps to minimize distortion while ensuring a sufficient quality of features. Despite the widespread use of advanced data generation mechanisms in NN-based MVI methods, such as autoencoders [9–12], generative adversarial networks (GANs) [13–16], and attention mechanisms [17–19], the aforementioned chain approach, although widely used in multilabel learning (e.g., classifier chains), has been rarely explored in NN-based MVI studies.

In this study, we focus on an effective MVI method following asymmetric strategy. Specifically, the proposed method first identifies the features without missing values from the original dataset. Then, the employed NN is trained on the selected features, considering the next feature with missing values as a target. After the target feature is imputed, it is added to the input of the subsequent imputation steps. The experimental results and statistical tests on 25 publicly available datasets showed that the proposed method significantly outperforms the conventional methods regarding missing value prediction accuracy. The implementation is accessible via GitHub at <https://github.com/KhrTim/ChainImputer>, accessed on 18 May 2025, supporting reproducibility and further research.

The remainder of this paper is organized as follows. Section 2 reviews existing approaches to missing value imputation, including classical, neural, attention-based, and diffusion-based methods. Furthermore, we describe our proposed solution and provide

the mathematical foundation and time complexity analysis in Section 3. Section 4 contains an experimental evaluation of our method in comparison with counterparts. Finally, in Section 5, we provide additional experimental results comparing variations of the proposed method.

2. Related Work

As MVI is a steady topic in machine learning, statistical imputation methods are widely used for handling missing data. Figure 1 shows a taxonomy of the imputation methods. For example, in the work of [20], statistical methods are chosen for their simplicity. Mean, mode, and median imputations are simple techniques that fill in missing values by computing the corresponding statistic from the observed values within each feature. Statistical imputation methods offer the advantage of being fast and straightforward in imputing missing values. However, the drawback is that they tend to underestimate variance and ignore relationships with other variables, making it challenging to impute accurate values [5].

Machine learning-based imputation involves constructing predictive models to estimate missing values in a dataset [21]. One of the advantages of machine learning-based imputation is its significantly greater flexibility compared to statistical methods, enabling better predictions by capturing higher-order interactions within the data [5]. The k -nearest neighbor (k -NN) imputation is one type of machine learning-based imputation methods [22]. k -NN imputation replaces missing values using information from the k -nearest neighboring data points. This enhances the completeness of the dataset while maintaining the valuable information for further analysis. However, inaccuracies may occur if the missing feature patterns are complex or unique. Additionally, locating neighbors for imputation can be challenging if the data distribution is highly imbalanced.

There are many ongoing studies related to machine learning-based imputation [23]. The experimental studies demonstrate the validity of various machine learning-based MVI methods [21]. While the chain approach has been widely applied in various domains, such as classifier chains in multilabel learning [24], the first imputation method based on this approach was proposed by Van Buuren and Groothuis-Oudshoorn [25], introducing the widely used Multiple Imputation by Chained Equations (MICE). Conventionally, Tsai C. et al. [26] conducted a comparative performance experiment among five widely used supervised learning methods: k -nearest neighbors (k -NN), classification and regression trees (CART), multi-layer perceptron neural networks (MLP), Naive Bayes, and support vector machines (SVMs). In their study, Yadav et al. [27] compared the experimental results of missing value imputation using visualization and imputation of missing values (VIM), MICE, nonparametric missing value imputation using random forest (MissForest), and Harrell miscellaneous (HMISC) methods. Wang et al. [28] performed a comparative analysis using two renowned imputation methods—autoregressive integrated moving average (ARIMA) and linear interpolation (LI) models—as well as three machine learning approaches: k -NN, multi-layer perceptron (MLP), and support vector regression (SVR) (a Support Vector Machine (SVM) regression used for missing value imputation). Palaniv-inayagam et al. [29] analysed the performance of five machine learning models: Naive Bayes, SVM, k -nearest neighbours, random forest, and linear regression. The experimental results showed that the SVM classifier achieves the highest accuracy, highlighting its importance in diabetes research and addressing missing data. Li et al. [30] tested six machine learning algorithms to predict hospital readmissions within 30 days for elderly patients (aged 65 and above). Experiments were conducted using RandomForest, LogisticRegression, XGBoost, LGBM, MLP, and Random + XGBoost and the results indicated that the RF

+ XGBoost algorithm exhibited superior overall performance in terms of Area Under the Curve (AUC) for prediction.

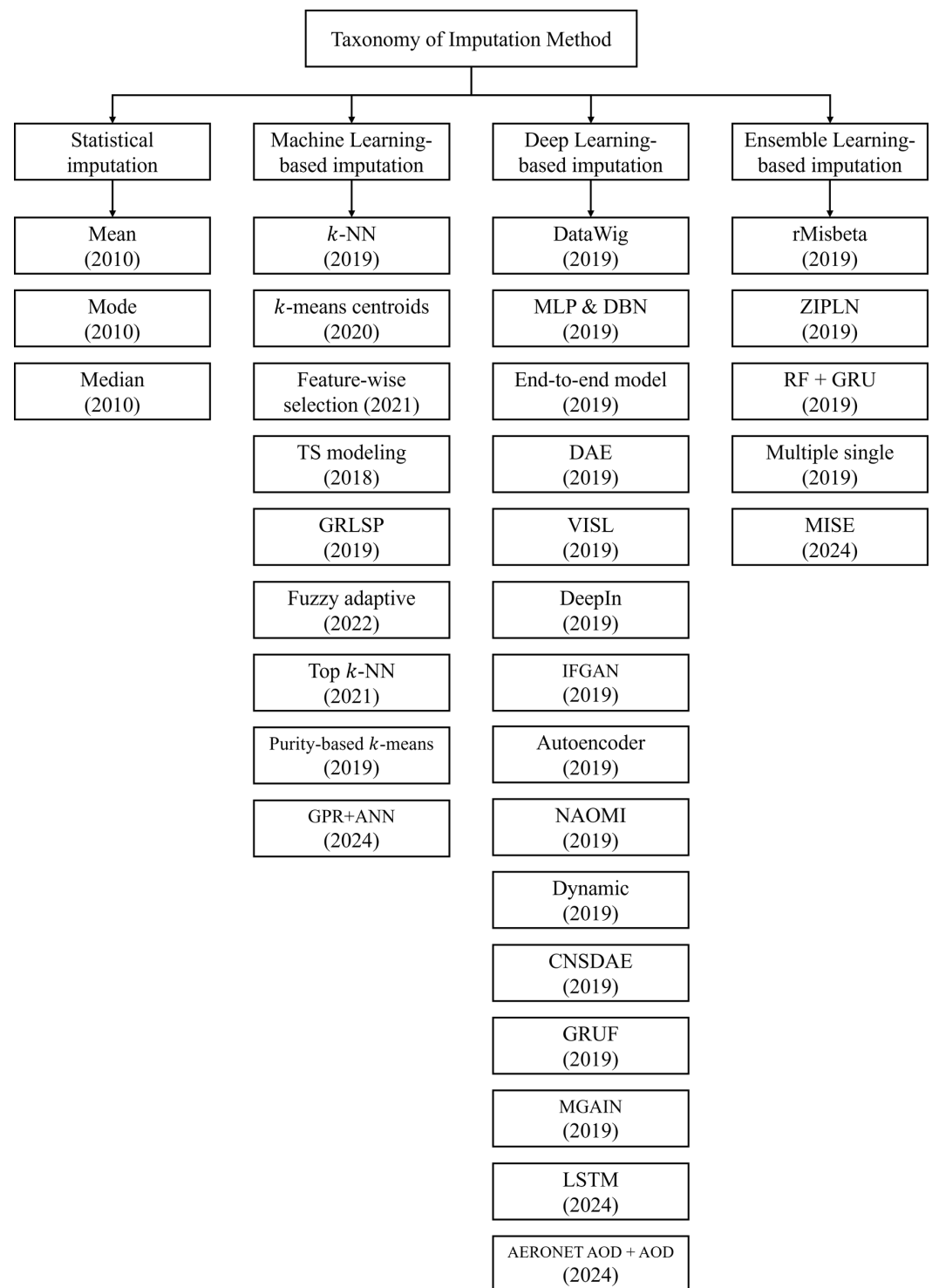


Figure 1. The taxonomy of imputation methods.

Raja et al. [31] introduced a novel imputation method based on rough k -means centroids that utilizes unsupervised machine learning methods to handle missing values. Another clustering-based method performs the imputation using a similarity-based spectral clustering approach, top k -nearest neighbor approach, and clustering [32]. Alternatively, the data-driven missing value imputation technique incorporates feature-wise selection to perform accurate imputations [33]. The Takagi–Sugeno (TS) modeling method is a

promising approach for incomplete datasets, utilizing Bayesian networks for missing value imputation [34]. The GRLSR method precisely imputes missing values in incomplete data by combining sample self-representation and local data structure [35]. Meanwhile, the fuzzy adaptive imputation approach (FAIA) is a fuzzy-based information decomposition method to address missing value problems in imbalanced data streams [36]. Another novel imputation approach is a query selection method that considers the imputation uncertainty in active learning with missing values [37]. In addition, the purity-based k -NN imputation method considers data purity and improves results by estimating high-purity instances as candidates [38]. Yet another K-NN method extension named Focalize K-NN leverages correlated features and temporal lags to improve the performance of the traditional K-NN imputer [39].

Deep learning-based imputation involves feeding data into the input layer of a neural network and computing the predicted output. The difference between each predicted value and its corresponding ground truth is then calculated [4]. In other words, training through multiple layers enables the model to discover complex relationships embedded within the data structure and improve imputation performance through fine-tuning [40]. Missing values can be imputed using traditional deep learning techniques such as long-short-term memory (LSTM) networks and Convolutional Neural Networks (CNN) [35]. A more recent type of deep learning-based imputation is DataWig [8]. DataWig (DW) creates a model for each feature to perform imputation. It aims to estimate the probabilities of possible values for a target feature using information from other features, including those with missing values. Accurately capturing the full range of data characteristics can be challenging in datasets with many missing values. As a result, the model may struggle to understand and handle missing values fully. Lin et al. [41] evaluated the performance of two supervised methods—Multi-layer perceptron (MLP) and deep belief networks (DBN). Morales-Alvarez et al. [42] introduced VISL—a novel scalable structure learning approach that simultaneously infers the structure between variable groups and imputes missing values using deep learning in the context of missing data. Another approach, DeepIn, is a missing value imputation solution for continuous missing patterns found in Internet-of-Things devices in smart spaces [43]. Alternatively, dynamic imputation is a method to enhance the training of neural networks in the presence of missing values [44]. Yet another DNN model imputes missing values in the hourly aerosol optical depth product by combining AERONET with a numerical model [45]. Furthermore, Bidirectional Recurrent Imputation for Time Series (BRITS) [46] treats missing values as trainable variables within a bidirectional recurrent neural network, allowing end-to-end optimization over time-series data.

The autoencoder introduces a new way to model the characteristics of a given dataset [47]. For example, an autoencoder can leverage information from neighboring data points to perform data imputation [9]. Expanding the applications of autoencoders, a different method was developed by modifying a denoising autoencoder (DAE) into a cluster-based imputation framework [10]. Gjørshoska et al. [11] utilized the autoencoder imputation method for resolving missing values in the food composition database (FCDB). A completely modified denoising stacking autoencoder (CMSDAE) is used for missing value imputation, particularly to enhance the quality of the MTL version [12]. Psychogyios et al. [48] proposed a method based on DAE with kNN for the pre-imputation task for the missing value imputation in the electronic health records.

An alternative solution comes from using the GAN, as was done by Zhang et al. [13], who applied the end-to-end GAN for multivariate time series, addressing the problem of imputed values being significantly different from actual values. Generative adversarial imputation networks (GAIN) [49] introduced a framework in which a generator imputes

missing values conditioned on observed data, while a discriminator learns to distinguish imputed from real values using a hint mechanism, achieving state-of-the-art results. Another GAN-based solution named IFGAN is a missing value imputation algorithm based on feature-specific GAN [14]. Furthermore, multiple generative adversarial imputation networks (MGAIN) is an imputation method that simplifies the network structure of GAIN and reduces the demand for data [15]. Non-autoregressive multi-resolution imputation (NAOMI) is a new deep generative model for imputing long-range sequences [16].

More recent studies explore the attention mechanism [50] capabilities in the MVI area tasks. One method of attention application was demonstrated by Koswar et al. [17], who proposed a framework that leverages between-feature or between-sample attentions. Another approach, named SAITS, operates with a weighted combination of two diagonally masked self-attention blocks, which explicitly capture temporal dependencies and feature correlations between time steps [18]. ImputeFormer combines strengths of low-rank and deep learning models, introducing a low rankness-induced transformer to balance strong inductive bias and high expressivity [19].

Another recent branch of MVI studies employs diffusion models for missing values generation. For example, Chen et al. apply a schrödinger bridge problem to probabilistic time series imputation by generating missing values conditioned on observed data [51]. Alternatively, in the work of Wang et al. [52], the evidence lower bound (ELBO) was re-derived in the scenario of multivariate time series imputation to take into account the correlations between observed and missing values. In the proposed multivariate imputation diffusion model (MIDM), the newly derived ELBO was enhanced with noise sampling and denoising mechanisms for multivariate time series imputation. Biloš et al. propose yet another method that adapts the noise generation and denoising mechanisms for time-series oriented scenario with irregularly sampled observations [53]. To overcome difficulties occurring in temporal electronic health records analysis, Dai et al. [54] proposed a Similarity-Aware Diffusion Model-Based Imputation (SADI), an imputation method that utilizes information across dependent variables. Following the diffusion ideas, Yang et al. [55] utilized a high-frequency filter to boost the residual term imputation, supplemented by a dominant-frequency filter for the trend and seasonal imputation in their proposed solution named multivariate time series imputation (FGTI). Lastly, to achieve an imputation consistency in terms of intra-consistency between observed and imputed values, and inter-consistency between adjacent windows, Zhou et al. [56] employed a contrastive complementary mask in Multivariate Time Series Consistent Imputation (MTSCI) to generate dual views during the forward noising process.

Ensemble-based imputation uses a combination of machine learning-based imputation and deep learning-based imputation. One type of ensemble-based imputation is stacked ensemble (SE) [57]. After filling the NaN values with the k -NN imputer, stacked ensemble (SE) creates a stacked ensemble classifier using models such as Extreme Gradient Boosting (XGB), random forest (RF), and Extra Tree Classifier (ETC). It is crucial to be aware that the predictions from each alternative imputation method are combined to make the final prediction. However, this approach can lead to a degradation in the performance of the final prediction if each model is overfitting.

Currently, there are many ongoing studies related to ensemble-based imputation. rMisbeta utilizes robust estimators based on the minimum beta divergence method. Experimental results show that data matrices imputed by rMisbeta outperform other statistical tools such as Zero, k -NN, SVD, EM, and RF. Furthermore, rMisbeta is an accurate, simple, and fast tool for missing value imputation [58]. Meanwhile, zero-inflated Poisson log-normal (ZIPLN) model is a feasible imputation method that uses a mixture distribution [59]. In particular, the missing data imputation method based on random forest proposes a

multi-step prediction for imputation and an ensemble model combining attention-based GRU models [60]. In addition, the multiple single imputation method based on ensemble learning applies missing value prediction using bootstrap sampling in the ensemble method, assigns weights to these predictions, aggregates them, and generates the final prediction [61]. Rao et al. proposed a multimodal imputation-based stacked ensemble (MISE) model to classify and predict air quality [62]. It has demonstrated superior performance through experimental validation. Jung et al. [63] utilized a bagging ensemble of multi-layer perceptrons, known as a Softmax ensemble network, to determine the ensemble weights for each MLP. Samad et al. [64] demonstrated that the linear regressor of MICE, replaced by ensemble learning and deep neural networks, improves both the imputation accuracy of MICE and the classification accuracy of imputed data.

As missing value imputation methods continue to evolve, increasing emphasis is being placed on their applicability to real-world scenarios, where data may be irregular, incomplete, and collected under complex conditions. For instance, GRUF captures the past states of time series data collected from devices and appropriately fills in missing values using both historical sensor readings and temporally aligned data from neighboring nodes via edge computing [65]. A recent study by Jiang and Zhang [66] proposed an interpretable diffusion-based imputation framework tailored for industrial soft sensing, incorporating resampling strategies and Fourier-based components to enhance both accuracy and model transparency.

3. Proposed Method

Table 1 summarizes the notations used to describe the proposed method. F denotes a set of datasets, comprising an input vector x_i and a corresponding label y_i , represented as $F = (x_i, y_i)_{i=1}^f$. Here, f denotes a feature, while each observation in the dataset consists of multiple such features. MVI aims to fill in missing values within the dataset ($F_{missing}$). It selects input features based on the provided list of features (train_col) and trains an imputation model denoted as M to predict missing values using the selected input features. After filling in missing values with the model M , denoted as $F_{imputed}$, the dataset is updated.

Table 1. Notations used for describing the proposed method.

Symbol	Name	Description
F	Dataset	Entire dataset
x	Input	Input features or independent variables
y	Output	Output or dependent variable
f	Number of features	Total number of features
$F_{missing}$	Dataset with missing values	Dataset containing missing values
$F_{imputed}$	Dataset by imputing missing values	Datasets without missing values
M	Imputation model	Model used for imputation

The procedure of conventional NN-based MVI methods is as follows [8]. Suppose we have an original feature set F where its features may contain missing values. Then, the algorithm chooses a target feature f containing missing values from F . Next, all the other features $F \setminus f$ are chosen to form the training dataset for NN. After that, a naïve MVI, such as mean imputation, is performed on each feature in $F \setminus f$ if it contains missing values. Thus, $F \setminus f$ can be filled with many roughly guessed values, resulting in an inaccurate prediction model. Lastly, $\hat{f}$, which is an imputed version of f by trained NN, replaces f from F , and then these steps are repeated until there are no features with missing values in F . We argue that the advantage of employing NN, which is expected to yield accurate predictions on the missing value based on its strong fitting capability, can be nullified because NN will learn the roughly guessed values for its prediction at all repetitions. Instead, we may consider

a natural imputation strategy as follows. Starting from F , the algorithm chooses f_1 with missing values and tries to train an NN using features $F \setminus f_1$. Because the training process cannot proceed if there are missing values in $F \setminus f_1$, the algorithm chooses f_2 , which is the second target feature of imputation, and then tries to train an NN using features $F \setminus \{f_1, f_2\}$. At the end of this recursive procedure, a set of features will remain that contains no missing values. Hence, a new feature, a carefully imputed version of the original feature, can be obtained after the employed NN is trained and imputation is performed. Next, in the backward stage of the recursive procedure, these features are inherently involved in the training of NN for the current recursion. When the recursion process is completed, all the missing values in F are imputed.

In Figure 2, the proposed method models for each feature using the available data. When missing values are encountered in the first feature, a common approach consists of replacing them with the mean of the existing values within the feature because there are no trainable features without missing values. If there are features without missing values, an NN can be used to train on those features and then impute the missing value of the first feature. Next, the missing values in the second feature are imputed based on the NN trained on the first feature. This iterative process continues sequentially for each feature. It allows for the gradual refinement of the models and incorporates more complex relationships between features. We argue that this iterative approach helps reduce the impact of missing values and adjusts the imputation process according to the characteristics and dependencies present in each dataset feature.

In Figure 3, the proposed method iteratively trains models to handle missing values, utilizing previously imputed missing values to train each new model and replacing all missing values in the dataset. The proposed method trains a model using the current input features and leverages it to replace missing values, ensuring a robust and reliable data analysis process. The training process of the proposed method involves constructing independent models for each feature and using the imputed results from the preceding feature to predict missing values sequentially. This process can be expressed as follows. First, the training for the first feature is conducted as

$$M_1 = M(F_{\text{imputed}_1}, F_{\text{imputed}_1}^{(1)}), \quad (1)$$

where F_{imputed_1} represents the dataset after filling in the missing values for the first feature. Subsequently, the training for the second feature is carried out as

$$M_2 = M(F_{\text{imputed}_{1,2}}, F_{\text{imputed}_{1,2}}^{(2)}), \quad (2)$$

where $F_{\text{imputed}_{1,2}}$ denotes the dataset after filling in missing values for both the first and second features, and $F_{\text{imputed}_{1,2}}^{(2)}$ represents the dataset used to train the model for the second feature. This iterative process continues for each subsequent feature. Specifically, the model M_i for the i -th feature is trained using the imputed results from features 1 through i . This can be written as

$$M_i = M(F_{\text{imputed}_{1,\dots,i}}, F_{\text{imputed}_{1,\dots,i}}^{(i)}), \quad (3)$$

where $F_{\text{imputed}_{1,\dots,i}}$ represents the dataset after filling in missing values for features 1 through i , and $F_{\text{imputed}_{1,\dots,i}}^{(i)}$ represents the dataset used to train the model for the i -th feature. This process ensures that each feature-wise model is trained using the imputed results from previous features, facilitating accurate imputation of missing values across the dataset.

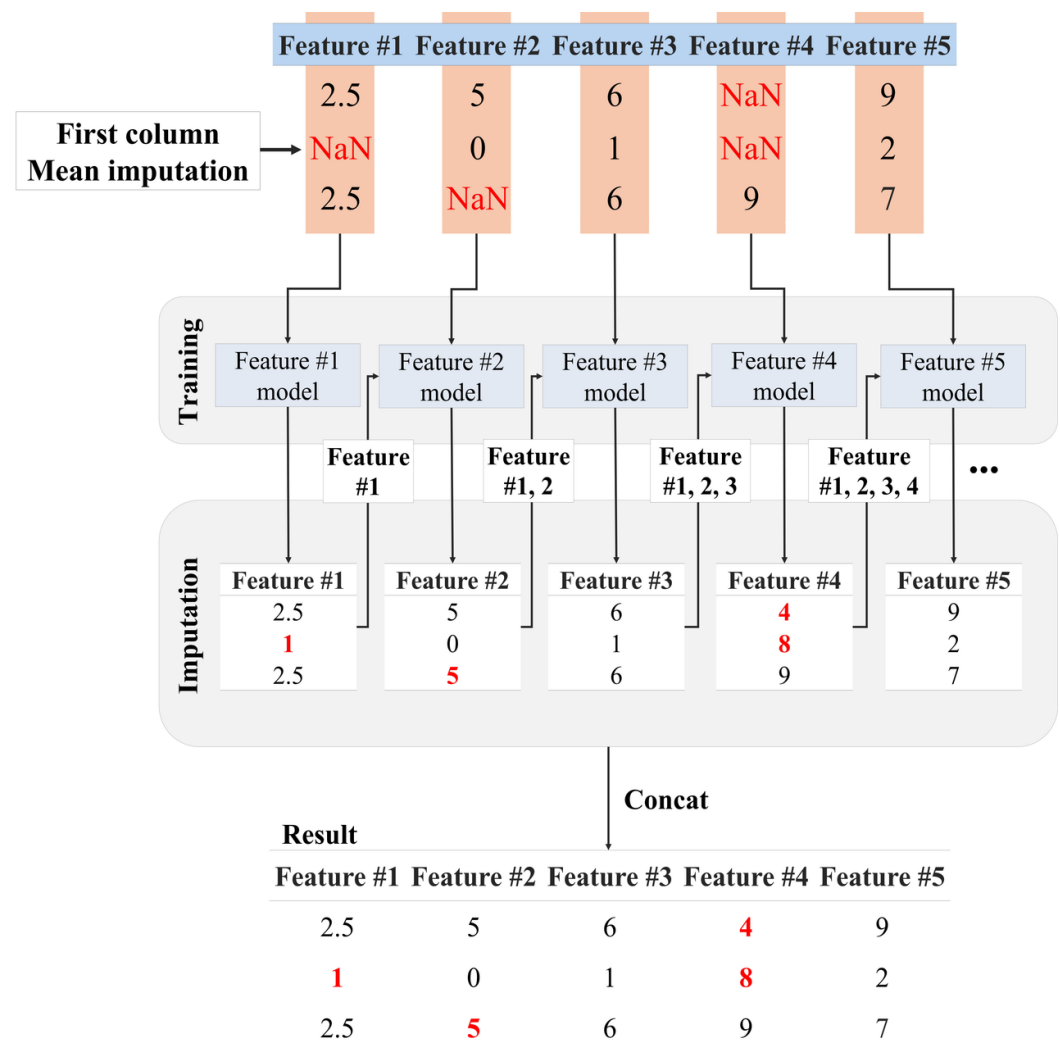


Figure 2. An illustrated example of the proposed iterative imputation method for handling missing data across multiple features. The process begins with mean imputation for the first feature column to initialize missing values. Each feature with missing values is then predicted using a model trained on the other available features. Specifically, Feature #1 is imputed using its mean, while subsequent features are imputed using models trained on the observed values of previously imputed features. The training step shows models trained with incrementally more input features (e.g., Feature #2 model is trained on Feature #1, Feature #3 model is trained on Features #1 and #2, etc.). The imputation step then applies these models to fill in missing values. Finally, the imputed features are concatenated to produce a fully imputed dataset. Red numbers in the imputation table and result indicate values that were originally missing and subsequently imputed by the proposed method.

It is worth noting that, unlike traditional chain-based methods such as MICE, the proposed method introduces two key differences. First, the feature ordering for imputation is not determined randomly, but is predefined based on the ascending order of missingness ratios. Second, the proposed method adopts an asymmetric structure by incrementally constructing a cumulative feature set. Each newly imputed feature is incorporated into the input of the subsequent imputation step, enabling more informed training and progressive refinement of the neural models. This progressive structure helps reduce the distortion caused by unreliable inputs in the early stages of the imputation process.

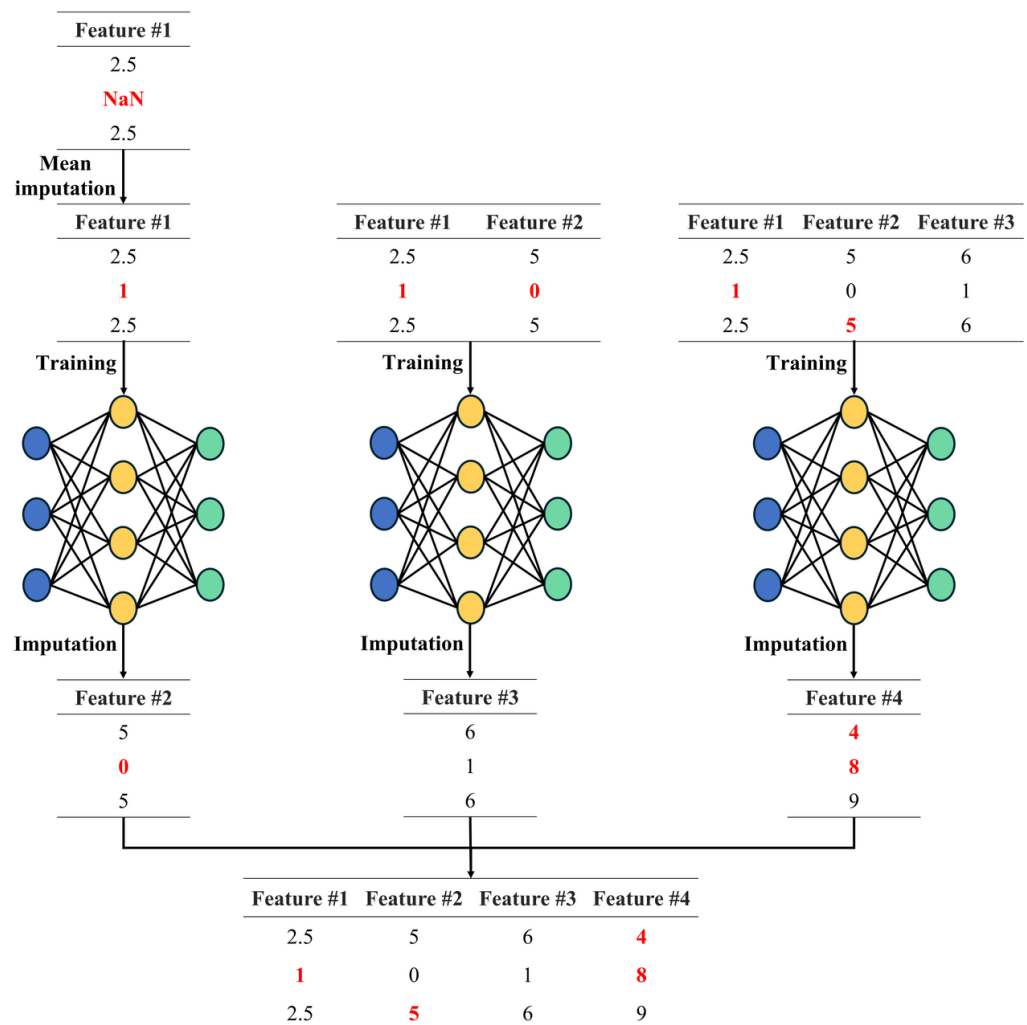


Figure 3. A step-by-step numerical example illustrating the proposed deep learning-based imputation strategy. The process begins with mean imputation for Feature #1 to handle missing values and provide initial inputs. Then, a neural network is trained to predict each subsequent feature using the previously imputed features as inputs. For example, Feature #2 is predicted using Feature #1, Feature #3 is predicted using Features #1 and #2, and so on. Each model is trained only on the rows without missing values for the target feature. During imputation, the trained model is used to fill in missing values. Red text highlights the originally missing entries that were imputed by the network. The final concatenated result presents the complete dataset with all missing values filled.

Algorithm 1 presents a method for imputing missing values in a dataset. The proposed method, called ChainImputer, employs a chain approach, leveraging initially complete features or mean imputation, followed by model-based imputation using MXNet. First, the original feature set F is initialized (Line 2), and S , which will contain features without missing values, is set to an empty set (Line 3). The algorithm iterates over each feature f in the feature set F (Line 4). For each feature, it checks if there are missing values (Line 5). If a feature f has no missing values, it is added to the set S (Line 6). If, after processing all features, the set S is empty, indicating that all features in F have missing values, the algorithm selects a random feature f from the feature set F (Line 10). It then uses a mean imputation method to fill in the missing values of f to create the initial feature (Line 11). The imputed feature f' is then added to the set S and removed from F (Lines 12–13). For each remaining feature f in the feature set F , the algorithm uses an MXNet model trained on the features in S to impute the missing values in f (Line 16). Next, the original feature

f is removed from the feature set F , and the imputed feature f' is then added to the set S (Lines 17–18). This process is repeated until all the features in F are considered.

Algorithm 1 Procedures of proposed imputation method

```

1: procedure CHAINIMPUTER( $F$ )                                ▷ Proposed missing value imputer
2:   Initialize  $F$  to the original feature set;
3:    $S \leftarrow \emptyset$ ;
4:   for each feature  $f \in F$  do
5:     if  $f$  has no missing values then
6:        $S \leftarrow S \cup f$ ;
7:     end if
8:   end for
9:   if  $S = \emptyset$  then                                       ▷ When there is no features for training MXNet
10:    Choose a random feature  $f \in F$ ;
11:     $f' \leftarrow \text{meanImputer}(f)$ ;
12:     $F \leftarrow F \setminus f$ ;
13:     $S \leftarrow S \cup f$ ;
14:  end if
15:  for each feature  $f \in F$  do
16:     $f' \leftarrow \text{mxNet}(S, f)$ ;                               ▷ complete  $f$  using MXNet trained based on  $S$ 
17:     $F = F \setminus f$ ;
18:     $S \leftarrow S \cup f'$ ;
19:  end for
20: end procedure

```

An imputed entry is an estimate and may deviate from the true value. If that entry is repeatedly reused as an input during later stages of the chain, its estimation error can propagate and accumulate, potentially degrading subsequent predictions. Therefore, an imputation schedule that minimizes the number of times previously imputed entries are reused provides a principled way to limit cumulative noise. We now show that the ascending-missingness order achieves this minimum.

Let $n_1^m \leq n_2^m \leq \dots \leq n_{|F|}^m$ denote the missing-value counts sorted in ascending order. If the feature with n_i^m missing entries is imputed at position i , its column will be reused in exactly $|F| - i$ subsequent models. The total number of imputed entries reused for an ordering π is therefore formalized as follows:

Proposition 1 (Re-use cost). *For any permutation π of $\{1, \dots, |F|\}$, the cumulative count of imputed entries that are reused during the chain is*

$$T(\pi) = \sum_{i=1}^{|F|} n_{\pi(i)}^m (|F| - i). \quad (4)$$

Lemma 1. *Consider two positions, $i < j$ with $n_i^m < n_j^m$, placed in reverse order inside π . Swapping their positions decreases $T(\pi)$.*

Proof. Let $d_i = |F| - i$ and $d_j = |F| - j$ so that $d_i > d_j$. Before the swap, the contribution of these two features to (4) is $n_j^m d_i + n_i^m d_j$. After swapping, it becomes $n_i^m d_i + n_j^m d_j$. Their difference satisfies

$$(n_j^m - n_i^m)(d_i - d_j) > 0, \quad (5)$$

so the swap strictly reduces $T(\pi)$. $\square$

Theorem 1 (Optimality of ascending order). *The ascending-missingness schedule $n_1^m, n_2^m, \dots, n_{|F|}^m$ minimizes $T(\pi)$ in (4) over all permutations π .*

Proof. Starting from any permutation, repeatedly apply the swap described in Lemma 1 to every inverted pair. Each swap lowers $T(\pi)$ and strictly reduces the inversion count. The process terminates only when no inversions remain, i.e., when the list is in ascending order, which therefore attains the global minimum. $\square$

By Theorem 1, the ascending-missingness order minimizes $T(\pi)$; hence, it reuses the fewest imputed entries and introduces the least cumulative distortion in subsequent training steps.

Based on Algorithm 1, we analyze the time complexity of the proposed ChainImputer and compare it with the neural network-based baseline DW for a balanced perspective. Let n denote the number of instances, $d = |F|$ the number of features, a the overall missing rate, T the number of training epochs, $g(n, a)$ the cost of applying a simple filler to one column (for example, mean imputation so that $g(n, a) = \mathcal{O}(n)$); and $h(n)$ the cost of one network epoch over the entire data set. DW first imputes the complete table once and then trains for T epochs [8], which leads to

$$\mathcal{O}(g(n, a) + T h(n)). \quad (6)$$

ChainImputer processes the d columns in sequence: for each feature it produces predicted values for that column and then trains the network for T epochs, resulting in

$$\mathcal{O}(d g(n, a) + d T h(n)). \quad (7)$$

Although this worst-case big- $\mathcal{O}$ bound is larger, ChainImputer begins with a single input feature, and the effective $h(n)$ increases gradually as the model incorporates more columns during the imputation process. Consequently, for data sets with substantial sparsity or moderate feature counts, its observed running time can be comparable to, or even less than, that of DW in practice, despite the higher asymptotic bound.

4. Experimental Results

This section presents experimental comparisons between the proposed model and existing approaches to validate its performance. We describe the experimental design to support this evaluation, covering the dataset, baseline imputation models, performance metrics, and statistical testing procedures.

4.1. Experimental Settings

The UCI machine learning repository dataset is used for the benchmark. The datasets represent various domains: health, social, biology, business, etc. Table 2 provides descriptions of the datasets used in the experiments. The datasets employed contain attributes and instances across a diverse range. The number of classes remains consistent for binary classification. A detailed description of employed datasets can be found in Appendix A. These datasets, each with their unique characteristics, serve as valuable resources for evaluating the effectiveness of the proposed imputation method across diverse domains and data types. Each dataset was randomly corrupted with missing values at a rate of 20% of the original data for the experiment; therefore, each prepared dataset falls under the Missing Completely At Random (MCAR) category. This preprocessing step was conducted before the experiments to compare the performance of missing value imputation methods. A 30-fold random split was applied to datasets, with each split following an 8:2 ratio for training and testing.

Table 2. Summary of datasets used in the experimental evaluation of the imputation methods. Each dataset is characterized by its domain, number of instances (rows), number of attributes (columns), total number of missing values, and data types (categorical, integer, or real-valued). These datasets span various fields including biology, medicine, business, and social science, ensuring the generalizability and robustness of the proposed method. Dataset references are provided for reproducibility.

Dataset	Domain	Instances	Attributes	Missing Values	Type	Reference
Abalone	Biology	4176	9	7517	Categorical	[67]
Adult	Social	32,560	15	97,680	Categorical	[68]
Ai4i	Computer Science	10,000	6	12,000	Real	[69]
Bank-marketing	Business	4521	17	15,371	Categorical	[70]
Blood	Business	748	5	748	Real	[71]
Breast	Health, Medicine	285	10	570	Categorical	[23]
Contraceptive	Health, Medicine	1473	10	2946	Categorical, Integer	[72]
Credit	Business	689	16	2205	Integer, Real	[73]
Echocardiogram	Health, Medicine	129	12	310	Integer, Real	[74]
Forty	Social	320	10	640	Integer, Real	[75]
Haberman	Health, Medicine	305	4	244	Integer	[76]
Heart	Health, Medicine	302	14	846	Categorical	[23]
Hepatitis	Health, Medicine	154	20	616	Integer, Real	[77]
Iris	Biology	149	5	149	Real	[78]
Liver	Health, Medicine	344	7	482	Categorical	[79]
Lymph	Health, Medicine	148	19	562	Categorical	[67]
Magic	Physics, Chemistry	19,019	11	41,842	Real	[80]
Maternal	Health, Medicine	1014	7	1420	Real	[81]
Monk	Social	415	7	581	Categorical	[82]
National-health	Health, Medicine	2278	9	4101	Real	[83]
Obesity	Health, Medicine	2111	17	7191	Integer	[84]
Rice	Biology	3809	8	6094	Real	[85]
Wholesale	Business	440	7	616	Integer	[86]
Wine	Social	177	14	496	Integer, Real	[34]
Yeast	Biology	1484	9	2671	Real	[87]

The proposed model compared three models using *RMSE*. The parameters for the three comparison models were set according to the values recommended in their respective studies. The description of the three comparative experimental models is as follows:

- DataWig (DW): DW creates a model for each feature to perform imputation. It aims to estimate the probability of all potential values for a feature given an imputation model and information from other features, especially those containing NaN values [8].
- *k*-Nearest Neighbor (*k*-NN) imputation: *k*-NN imputation replaces missing values by using the nearest neighbor observation value *k* for data points with missing values. This enhances the completeness of the dataset and allows for valuable information for analysis [22].
- Stacked Ensemble (SE): After filling the NaN value with the *k*-NN imputer, it predicts a stacked ensemble classifier with Extreme Gradient Boosting (XGB), random forest (RF), and Extra Tree Classifier (ETC) models [57].

Specifically, DW was chosen because it has been shown to outperform GAIN, a GAN-based method, and DAE in its original study [8]. SE integrates a variety of traditional machine learning algorithms, including tree-based models, and has demonstrated superior performance over other conventional imputation techniques. Therefore, DW and SE offer strong baseline comparisons from both deep learning and traditional machine learning perspectives.

Table 3 shows the hyperparameter settings of each method where that of the comparison models were referenced from [44]. In contrast, the hyperparameters for the proposed

method were determined based on its rapid convergence in preliminary experiments. In particular, we employed a regression loss with ReLU activation to better suit the continuous nature of the imputation task, and used *RMSE* as the evaluation metric, which is standard for assessing numerical reconstruction accuracy. A neural network with one hidden layer was constructed using a TensorFlow-based framework. The use of a single hidden layer is motivated by the incremental nature of the chain imputation process, where the number of input features is initially small and increases gradually. In such settings, deeper architectures may lead to overfitting, especially during the early stages of training. Similarly, the proposed method utilized an MXNET-based framework to create a neural network with one hidden layer, consistent with the comparison experimental model [8]. Specifically, MXNET was chosen for its lightweight symbolic-imperative hybrid engine, which enabled rapid prototyping with lower GPU memory consumption. *RMSE* is employed as the evaluation measure. The metric is the most commonly used metric to estimate the performance of the missing value imputation methods [88]. It measures the difference between imputed values and actual values. After averaging the squared errors between predicted and actual values, the square root of this value is taken to calculate $RMSE = \sqrt{\frac{\sum_{i=1}^n (y - \hat{y})^2}{n}}$, where y and $\hat{y}$ represent the original value, but hidden to the MVI methods, and the imputed data, respectively. A lower value indicates higher performance.

Table 3. Hyperparameter settings used in our experiments.

Method	Batch Size	Epochs	Loss Function	Optimizer	Learning Rate
Proposed	8	10	Regression	<i>ReLU</i>	4×10^{-3}
DW	32	50	Cross-entropy	<i>Adam</i>	1×10^{-3}
<i>k</i> -NN	32	50	Cross-entropy	<i>Adam</i>	1×10^{-3}
SE	32	50	Cross-entropy	<i>Adam</i>	1×10^{-3}

Lastly, statistical tests are performed to validate the superiority among imputation models. The Friedman test is a non-parametric equivalent that assesses repeatedly measured outcomes by separating the dataset by the algorithm and assigning ranks to measure them [89], written as

$$R_j = \frac{1}{N} \sum_{i=1}^N r_i^j.$$

The rank of the j -th method for the i -th dataset, derived from the experimental results, can be denoted as r_i^j when employing N datasets and k methods. Assuming equality of R_j , χ_F^2 is calculated as

$$\chi_F^2 = \frac{12N}{k(k+1)} \left(\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right).$$

Then, Friedman statistics F_F are distributed according to the F_F -distribution, with $k - 1$ and $(k - 1)(N - 1)$ degrees of freedom, that can be calculated as

$$F_F = \frac{(N - 1)\chi_F^2}{N(k - 1) - \chi_F^2}.$$

If there is a statistically significant difference ($p < 0.05$), a post hoc test is conducted in the Friedman test. The Bonferroni–Dunn test was chosen as the post hoc test because there was a statistically significant difference between this and the Friedman tests. The Critical Difference (CD) is used to compare the differences between the average ranks of algorithms, written as

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}.$$

4.2. Comparison Results and Analysis

Table 4 presents experimental results for employed datasets, showing the performance of the proposed method alongside three comparison models, focusing on the *RMSE* values. These results indicate the robustness and efficacy of our proposed method in accurately imputing missing values. The experimental results, represented in average *RMSE* and standard deviation, underscore the superiority of the proposed method over the comparison models. The proposed method achieves superior results on 22 out of the 25 datasets compared to existing methods. Notably, on the Abalone dataset, the proposed method achieved a performance score of 0.2007, clearly outperforming the comparison models. Similarly, the proposed method demonstrated superior performance in the Adult dataset with a mean *RMSE* of 0.1894 compared to the other models. In addition, the experimental results indicate that the proposed method exhibits remarkable consistency and effectiveness across various datasets. For instance, it consistently outperforms the comparison models in datasets such as Blood, Contraceptive, and Magic, as evidenced by its significantly lower *RMSE* values. With an average rank of 1.24, the method consistently proved its effectiveness.

Table 4. Root Mean Square Error (*RMSE*) comparison across datasets for the proposed imputation method and three baselines: DW (Datawig), *k*-NN (K-nearest neighbors), and SE (Stacked Ensemble). The bottom row shows the average rank of each method across all datasets, where a lower rank indicates better overall performance. The proposed method achieves the lowest average rank (1.24), demonstrating superior imputation accuracy across diverse datasets.

Dataset	Proposed	DW	<i>k</i> -NN	SE
Abalone	0.2007 ± 0.0599	0.5095 ± 0.0100	0.5531 ± 0.0103	0.6562 ± 0.0025
Adult	0.1894 ± 0.0946	0.2146 ± 0.0011	0.2582 ± 0.0006	0.5803 ± 0.0010
Ai4i	0.1658 ± 0.0401	0.1724 ± 0.0009	0.2448 ± 0.0037	0.3185 ± 0.0016
Bank-marketing	0.2209 ± 0.1340	0.2621 ± 0.0013	0.3186 ± 0.0014	0.3842 ± 0.0030
Blood	0.1573 ± 0.0648	0.2227 ± 0.0492	0.2670 ± 0.0486	0.5643 ± 0.0268
Breast	0.3897 ± 0.1880	0.5641 ± 0.0153	0.7109 ± 0.0168	0.6371 ± 0.0122
Contraceptive	0.2815 ± 0.0736	0.2884 ± 0.0017	0.3589 ± 0.0026	0.6943 ± 0.0042
Credit	0.2836 ± 0.1344	0.3368 ± 0.0321	0.3905 ± 0.0308	0.7025 ± 0.0205
Echocardiogram	0.3054 ± 0.1292	0.3853 ± 0.0461	0.4030 ± 0.0525	0.5213 ± 0.0716
Forty	0.2238 ± 0.0686	0.2602 ± 0.0280	0.3151 ± 0.0271	1.2588 ± 0.0478
Haberman	0.1892 ± 0.0689	0.2222 ± 0.0288	0.3045 ± 0.0543	1.2584 ± 0.0584
Heart	0.3094 ± 0.1359	0.3699 ± 0.0043	0.4442 ± 0.0083	0.7026 ± 0.0098
Hepatitis	0.3547 ± 0.1433	0.4654 ± 0.0380	0.4182 ± 0.0334	0.9933 ± 0.0612
Iris	0.3267 ± 0.0619	0.5403 ± 0.0206	0.7072 ± 0.0363	0.6469 ± 0.0207
liver	0.1793 ± 0.1038	0.2610 ± 0.0047	0.3243 ± 0.0082	0.2460 ± 0.0115
Lymph	0.4645 ± 0.2037	0.4574 ± 0.0137	0.4302 ± 0.0112	0.7228 ± 0.0127
Magic	0.1316 ± 0.0631	0.1475 ± 0.0005	0.1693 ± 0.0009	0.6622 ± 0.0010
Maternal	0.3013 ± 0.1888	0.2543 ± 0.0125	0.2397 ± 0.0117	0.6346 ± 0.0059
Monk	0.3885 ± 0.0494	0.3885 ± 0.0122	0.5307 ± 0.0295	0.6863 ± 0.0358
National-health	0.1784 ± 0.1242	0.2783 ± 0.0103	0.3055 ± 0.0086	0.4986 ± 0.0026
Obesity	0.2380 ± 0.0769	0.2698 ± 0.0021	0.2819 ± 0.0024	0.4786 ± 0.0033
Rice	0.1755 ± 0.0315	0.1748 ± 0.0014	0.1883 ± 0.0019	0.7074 ± 0.0033
Wholesale	0.0722 ± 0.0111	0.1613 ± 0.0420	0.1726 ± 0.0448	1.2148 ± 0.0446
Wine	0.2339 ± 0.0673	0.2786 ± 0.0045	0.2589 ± 0.0072	0.6830 ± 0.0116
Yeast	0.1228 ± 0.0396	0.1641 ± 0.0236	0.1637 ± 0.0039	0.6207 ± 0.0031
Avg.Rank	1.24	2.08	2.84	3.84

As shown in Table 5, the Friedman test, conducted with a 95% confidence interval, confirmed statistically significant differences between the groups, with a Chi-squared statistic of 55.867 and a *p*-value of 2.276. Subsequently, the Bonferroni–Dunn test followed

as a post hoc test, revealing clear distinctions between the comparison models and the proposed method as shown in Figure 4. The proposed method consistently outperformed the comparison models, with a notable margin beyond $CD = 0.8742$. Experimental results showed the superior performance of the proposed method across 22 datasets, as evidenced by its significantly lower $RMSE$ values. Statistical tests further support the effectiveness of the proposed method, confirming its efficacy in accurately imputing missing values.

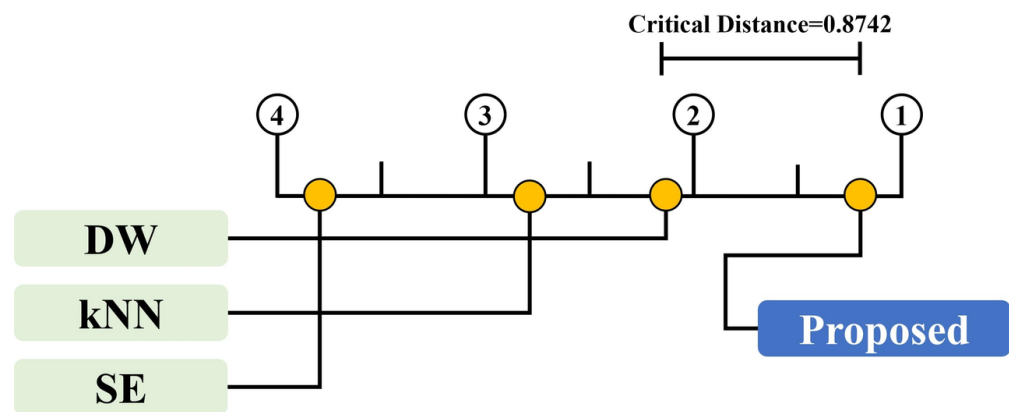


Figure 4. Result of Bonferroni–Dunn test between the proposed method and comparison models.

Table 5. Summary of the Friedman statistic $F_F(k = 4, N = 25)$ and Critical Value in terms of $RMSE$ measure between the proposed method and comparison models.

Evaluation Measure	Friedman Statistic	Critical Values ($\alpha = 0.05$)
$RMSE$	55.867	2.276

The proposed method uses a chain imputer, which sequentially constructs models for each feature while stacking them for training. Figure 5 illustrates the percentage change in $RMSE$ resulting from iterative feature refinement across different training iterations for two datasets: Breast and Yeast. For each dataset, a set of selected features is shown along the x-axis, while the y-axis indicates the relative change in $RMSE$ (%). The lines in each plot correspond to specific training iterations—namely, epochs 1, 20, 40, and 60—allowing observation of how the influence of individual features evolves as training progresses. The black dashed line at 0% serves as a baseline, denoting no change in $RMSE$. In the context of missing value imputation, a positive percent change implies that the permuted feature is important for accurate imputation—its disruption leads to increased reconstruction error. Conversely, a negative percent change suggests that the feature may be noisy or uninformative, as its permutation leads to a reduction in $RMSE$. Features near 0% are likely neutral, contributing minimally to the imputation process. From the plots, it can be observed that certain features, such as age and menopause in the Breast dataset and erl in the Yeast dataset, exhibit increasing positive influence on imputation accuracy across training iterations, indicating that the model increasingly relies on them. In contrast, features like inv-nodes and gvh show persistent negative impacts, possibly due to inconsistent or misleading patterns in the data. This analysis highlights how feature relevance for imputation evolves during training and can be used to assess feature robustness and model dependency in the presence of missing values.

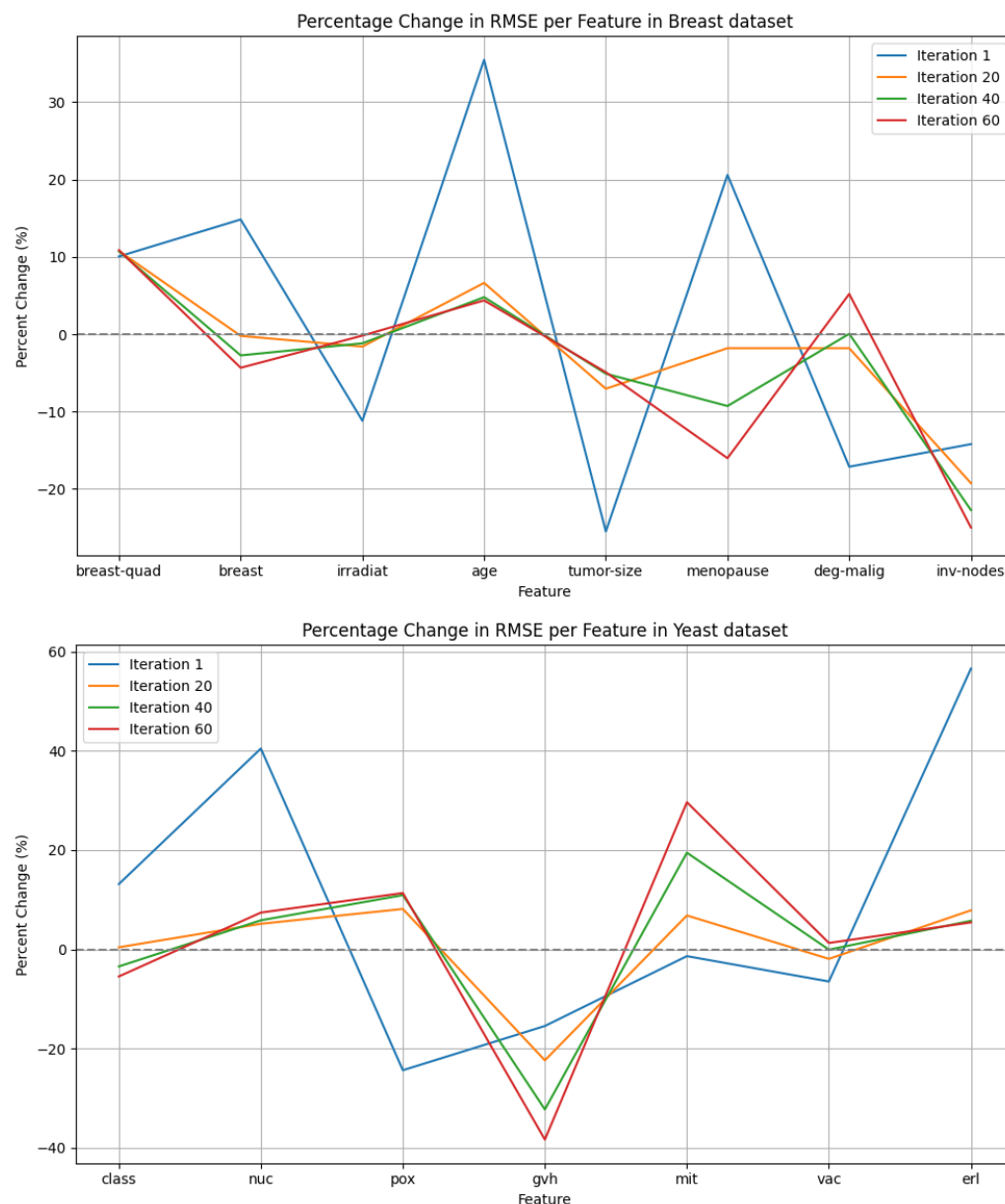


Figure 5. Percentage change in *RMSE* across training epochs (1, 20, 40, and 60) for selected features in the Breast and Yeast datasets. Positive values indicate important features whose permutation increases *RMSE*, while negative values suggest potentially noisy or uninformative features. The 0% dashed line denotes no impact. Feature relevance evolves over training, with age, menopause (Breast), and erl (Yeast) showing increasing importance, while inv-nodes and gvh exhibit consistent negative influence.

5. Discussion

In this section we discuss the results of additional experiments that affect our proposed solution performance. As shown in Table 6, our experiments have shown that the order in which features are selected for imputation significantly affects the overall accuracy of the value imputation process. Following the options of *IterativeImputer* class from the *scikit-learn* library, we conducted the experiments with three ordering schemes: ascending, descending, and random [90]. The ascending scheme is used in the default implementation of our method and, in this approach, the next column is selected if it contains the least amount of missing values among the unprocessed columns. For the descending iteration scheme, the next feature is chosen based on the maximal amount of missing values among unprocessed columns. Finally, in the random shuffle scheme, the

next feature is chosen randomly. In this experimental setup, each feature was individually corrupted with missing values ranging from 1% to 22%. This approach enables a clearer comparison of performance differences between ordering schemes, as opposed to scenarios where all features are corrupted at the same rate. Following our previous methodology, we conducted the experiments using 30-fold cross-validation. The results confirm that the default ordering scheme of the ChainImputer—ascending imputation—achieves the best performance. Processing features with fewer missing values first helps mitigate error propagation in subsequent features, leading to more accurate overall imputations.

ChainImputer can be extended to time-series or spatial domains by ordering features along temporal or spatial proximity, so early steps exploit the most reliable neighborhood while later steps refine heavily corrupted segments. In addition, even under severe missingness ($>50\%$), the proposed method retains all n instances at every step, whereas chained-equation baselines lose training rows as soon as a target column contains gaps. Because each iteration adds a full column of n newly imputed values, the effective sample size grows monotonically, supporting stable performance in high-dimensional settings with extreme data loss.

Table 6. RMSE comparison of the ChainImputer method under three different feature ordering schemes: ascending (ChainImputer), random shuffle, and descending order. The average rank across all datasets is presented in the final row, with a lower value indicating better overall performance. The default ordering scheme of ChainImputer consistently outperforms the other ordering strategies with the lowest average rank (1.64), suggesting that the ascending feature order provides more effective imputation performance than randomized or descending arrangements.

Dataset	ChainImputer	Random Shuffle	Descending
Abalone	0.2007 $\pm$ 0.0599	0.2028 $\pm$ 0.0614	0.2043 $\pm$ 0.0632
Adult	0.1894 $\pm$ 0.0946	0.1991 $\pm$ 0.0730	0.2096 $\pm$ 0.0186
Ai4i	0.1658 $\pm$ 0.0401	0.1771 $\pm$ 0.0075	0.1635 $\pm$ 0.0408
Bank-marketing	0.2209 $\pm$ 0.1340	0.2089 $\pm$ 0.1069	0.2193 $\pm$ 0.1352
Blood	0.1573 $\pm$ 0.0648	0.1591 $\pm$ 0.0645	0.1502 $\pm$ 0.0568
Breast	0.3897 $\pm$ 0.1880	0.3950 $\pm$ 0.1841	0.3925 $\pm$ 0.1789
Contraceptive	0.2815 $\pm$ 0.0736	0.2468 $\pm$ 0.0677	0.2658 $\pm$ 0.0632
Credit	0.2836 $\pm$ 0.1344	0.2840 $\pm$ 0.1342	0.2848 $\pm$ 0.1342
Echocardiogram	0.3054 $\pm$ 0.1292	0.3060 $\pm$ 0.1296	0.3057 $\pm$ 0.1297
Forty	0.2238 $\pm$ 0.0686	0.2431 $\pm$ 0.0379	0.1974 $\pm$ 0.0810
Haberman	0.1892 $\pm$ 0.0689	0.1800 $\pm$ 0.0731	0.2050 $\pm$ 0.0595
Heart	0.3094 $\pm$ 0.1359	0.3119 $\pm$ 0.1333	0.3121 $\pm$ 0.1325
Hepatitis	0.3547 $\pm$ 0.1433	0.3664 $\pm$ 0.1377	0.3869 $\pm$ 0.1290
Iris	0.3267 $\pm$ 0.0619	0.3330 $\pm$ 0.0624	0.3431 $\pm$ 0.0647
liver	0.1793 $\pm$ 0.1038	0.1990 $\pm$ 0.0595	0.1814 $\pm$ 0.1131
Lymph	0.4645 $\pm$ 0.2037	0.4673 $\pm$ 0.1977	0.4819 $\pm$ 0.2057
Magic	0.1316 $\pm$ 0.0631	0.1406 $\pm$ 0.0525	0.1484 $\pm$ 0.0157
Maternal	0.3013 $\pm$ 0.1888	0.3276 $\pm$ 0.1680	0.3111 $\pm$ 0.1804
Monk	0.3885 $\pm$ 0.0494	0.3893 $\pm$ 0.0502	0.3875 $\pm$ 0.0455
National-health	0.1784 $\pm$ 0.1242	0.1775 $\pm$ 0.2076	0.1958 $\pm$ 0.1155
Obesity	0.2380 $\pm$ 0.0769	0.2312 $\pm$ 0.0747	0.2453 $\pm$ 0.0751
Rice	0.1755 $\pm$ 0.0315	0.1584 $\pm$ 0.0380	0.1538 $\pm$ 0.0330
Wholesale	0.0722 $\pm$ 0.0111	0.0698 $\pm$ 0.0116	0.0705 $\pm$ 0.0115
Wine	0.2339 $\pm$ 0.0673	0.2348 $\pm$ 0.0674	0.2345 $\pm$ 0.0673
Yeast	0.1228 $\pm$ 0.0396	0.1224 $\pm$ 0.0390	0.1228 $\pm$ 0.0387
Avg.Rank	1.64	2.24	2.08

6. Conclusions

In this paper, the proposed chain imputer method not only demonstrates a systematic approach to handling missing values but also significantly enhances imputation accuracy. The results show the superiority of the proposed method in imputing missing values across diverse datasets. The proposed method proves to be both robust and effective, outperforming the comparison models by yielding lower *RMSE* values in 22 out of 25 datasets. Its high performance in datasets such as Abalone and Adult is particularly notable, suggesting that it can provide a reliable solution for imputing missing values across various fields. The Friedman and Bonferroni–Dunn post hoc tests further validate the superiority of the proposed method. The significant differences observed between the groups confirm that the proposed method consistently outperforms the comparison models. The Bonferroni–Dunn test highlights the clear superiority of the proposed method. Such statistical validation contributes to establishing the proposed method as a practical solution for MVI tasks and enhances confidence in the experimental results. In conclusion, the experimental results and in-depth analysis presented in this study suggest that by combining chain strategy and neural network modeling, a robust and adaptable methodology can ensure the reliability and performance of the data analysis process.

Despite its effectiveness, the proposed method has some limitations. First, the current implementation is tightly coupled with the MXNet framework, which may restrict portability and optimization opportunities in other environments. Future work will explore alternative architectures and platforms to improve scalability. Second, the imputation order, which is currently determined in a heuristic manner, can impact performance. A more principled approach based on entropy-driven feature informativeness will be investigated to derive an optimal imputation sequence. Third, the hyperparameters for each comparison method, including the proposed model, were not independently optimized but were adopted from previously published experimental settings. Although this ensures consistency and reproducibility, it may affect the comparative performance. As a future direction, systematic tuning procedures such as grid search or Bayesian optimization could be applied to each method to ensure a fair and balanced evaluation. These limitations point to important directions for future research that can further improve the robustness and generalizability of the proposed method.

Furthermore, valuable insights and outcomes are anticipated from applying the proposed method in various fields, potentially leading to advancements in research, industry, and decision-making practices. Enhancing imputation performance leads to more reliable results in various tasks such as predictive modeling, classification, and clustering. Imputation of missing values helps minimize the risk of distorted data and unreliable insights in data-driven analyses. Additionally, the advantage of utilizing the proposed method is its applicability to datasets with missing values across various fields. Extending this approach to larger and more diverse datasets, including real-world applications, is expected to produce reliable results.

Author Contributions: Conceptualization, W.S. and H.-J.B.; methodology, H.-J.B.; software, W.S. and H.-J.B.; validation, W.S., H.-J.B., and J.L.; writing—original draft preparation, W.S. and H.-J.B.; writing—review and editing, W.S. and T.K.; supervision, W.S. and J.L.; funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) [RS-2021-II211341, Artificial Intelligence Graduate School Program (Chung-Ang University)], and in part by the Chung-Ang University Young Scientist Scholarship in 2024.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found in the UCI Machine Learning Repository at <https://archive.ics.uci.edu> (accessed on 25 May 2025).

Conflicts of Interest: Author H.-J.B. is employed by the company Auton. Auton had no role in the design, execution, interpretation, or funding of this study. All authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Dataset Description

In this study, we employed 25 publicly available datasets. A detailed description of each dataset is provided as follows:

- **Abalone:** This dataset, originating from the field of biology, contains observations related to the age of abalone based on physical measurements such as diameter, height, and weight.
- **Adult:** In the realm of social sciences, the Adult dataset comprises demographic information such as age, education, marital status, and occupation, along with attributes related to income levels, specifically whether an individual earns more or less than USD 50,000 annually.
- **Ai4i:** The Ai4i dataset, which comes from computer science, provides insights into artificial intelligence and industrial automation. It offers various features related to manufacturing processes, such as temperature, pressure, and humidity, along with attributes indicating process parameters and yield.
- **Bank-marketing:** This dataset, sourced from business studies, includes information on marketing campaigns conducted by a Portuguese banking institution. It contains client demographics, contact methods, and campaign outcomes (e.g., whether a client subscribed to a term deposit).
- **Blood:** Also from the business domain, the Blood dataset contains data related to blood donation behavior. It includes attributes such as age, blood pressure, and donation frequency, aiding in understanding donor patterns and predicting donation likelihood.
- **Breast:** The Breast dataset presents attributes relevant to breast cancer diagnosis in the health and medicine domain. It includes features such as tumor size, lymph node status, and grade, facilitating the classification of tumors as benign or malignant.
- **Contraceptive:** Another health and medicine dataset, this one provides information on contraceptive method choices among women based on socioeconomic factors such as age, education level, and number of children. This dataset helps in studying contraceptive preferences and family planning behavior.
- **Credit:** This dataset encompasses credit card application data, offering insights into creditworthiness and risk assessment. It includes attributes such as credit score, income, and employment status, aiding in predicting credit approval or rejection.
- **Echocardiogram:** From health and medicine, this dataset contains echocardiographic measurements relevant for diagnosing heart conditions. Attributes include parameters such as ejection fraction, wall motion score, and presence of abnormalities, assisting in cardiac disease diagnosis and prognosis.
- **Forty:** In the social sciences domain, the Forty dataset offers attributes related to potential student performance predictors. It includes features such as student demographics, study habits, and socioeconomic background, aiding in identifying factors influencing academic success.
- **Haberman:** This dataset provides insights into the survival rates of patients who underwent breast cancer surgery. It includes attributes such as patient age, year of

operation, and number of positive axillary nodes detected, helping to analyze factors affecting survival after cancer treatment.

- **Heart:** Relevant to health and medicine, this dataset contains attributes helpful in predicting the presence of heart disease. It includes features such as age, cholesterol levels, and exercise-induced angina, facilitating cardiovascular risk assessment.
- **Hepatitis:** This dataset includes attributes related to hepatitis diagnosis, aiding in understanding the factors influencing liver health. It includes features such as patient demographics, laboratory test results, and history of alcohol consumption or drug use, assisting in hepatitis prognosis and treatment planning.
- **Iris:** The Iris dataset is a classic biology benchmark containing iris flower attributes for species classification. It includes features such as sepal and petal length and width, enabling the identification of iris species based on morphological characteristics.
- **Liver:** This dataset offers insights into liver health and disease diagnosis. It includes attributes such as patient demographics, liver function test results, and the presence of symptoms such as jaundice or ascites, aiding in the diagnosis of liver diseases such as hepatitis or cirrhosis.
- **Lymph:** This dataset, with attributes relevant to lymph node status, aids in understanding cancer prognosis. It includes features such as tumor size, grade, and lymph node involvement, facilitating the prediction of disease progression and treatment outcomes in cancer patients.
- **Magic:** This dataset provides observations related to high-energy gamma-ray sources in physics and chemistry. It includes attributes such as spectral and spatial characteristics of gamma-ray events, aiding in identifying and analyzing cosmic ray sources and astrophysical phenomena.
- **Maternal:** From health and medicine, this dataset offers insights into maternal health and pregnancy outcomes. It includes attributes such as maternal age, prenatal care, and birth outcomes (e.g., birth weight, gestational age), facilitating the study of factors influencing maternal and neonatal health.
- **Monk:** This dataset contains attributes relevant to predicting the outcome of a religious study. It includes features such as age, years of study, and spiritual practices, aiding in understanding factors influencing spiritual development and attainment.
- **National-health:** With attributes related to national health indicators, this dataset aids in understanding public health trends. It includes disease prevalence, healthcare expenditure, and population demographics, facilitating the analysis of health disparities and policy evaluation.
- **Obesity:** Relevant to health and medicine, this dataset offers insights into obesity prevalence and contributing factors. It includes attributes such as body mass index (BMI), dietary habits, and physical activity levels, aiding in studying obesity trends and risk factors for weight gain.
- **Rice:** This dataset provides attributes relevant to rice crop classification in biology. It includes features such as soil characteristics, climate conditions, and agricultural practices, aiding in analyzing factors influencing rice yield and quality.
- **Wholesale:** This dataset contains information on wholesale purchase behavior from the business domain. It includes attributes such as product categories, quantities purchased, and customer demographics, facilitating the analysis of purchasing patterns and market trends.
- **Wine:** In the social sciences domain, this dataset offers attributes related to wine quality assessment. It includes chemical composition (e.g., alcohol content, acidity), sensory properties (e.g., color, aroma), and wine ratings, aiding in understanding factors influencing wine quality and consumer preferences.

- Yeast: From biology, this dataset contains attributes relevant to yeast cell growth prediction. It includes genetic characteristics, growth conditions, and cellular responses to environmental stimuli, facilitating the study of yeast physiology and biotechnological applications.

References

1. Madhu, G.; Bharadwaj, B.L.; Nagachandrika, G.; Vardhan, K.S. A novel algorithm for missing data imputation on machine learning. In Proceedings of the 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 27–29 November 2019; pp. 173–177.
2. Aljuaid, T.; Sasi, S. Proper imputation techniques for missing values in data sets. In Proceedings of the 2016 International Conference on Data Science and Engineering (ICDSE), Cochin, India, 23–25 August 2016; pp. 1–5.
3. Awan, S.E.; Bennamoun, M.; Sohel, F.; Sanfilippo, F.; Dwivedi, G. A reinforcement learning-based approach for imputing missing data. *Neural Comput. Appl.* **2022**, *34*, 9701–9716. [\[CrossRef\]](#)
4. Lin, W.C.; Tsai, C.F. Missing value imputation: A review and analysis of the literature (2006–2017). *Artif. Intell. Rev.* **2020**, *53*, 1487–1509. [\[CrossRef\]](#)
5. Jerez, J.M.; Molina, I.; García-Laencina, P.J.; Alba, E.; Ribelles, N.; Martín, M.; Franco, L. Missing data imputation using statistical and machine learning methods in a real breast cancer problem. *Artif. Intell. Med.* **2010**, *50*, 105–115. [\[CrossRef\]](#)
6. Pham, T.M.; Pandis, N.; White, I.R. Missing data: Issues, concepts, methods. In *Seminars in Orthodontics*; Elsevier: Amsterdam, The Netherlands, 2024; Volume 30, pp. 37–44.
7. Whitehead, T.M.; Irwin, B.W.; Hunt, P.; Segall, M.D.; Conduit, G.J. Imputation of assay bioactivity data using deep learning. *J. Chem. Inf. Model.* **2019**, *59*, 1197–1204. [\[CrossRef\]](#)
8. Biessmann, F.; Rukat, T.; Schmidt, P.; Naidu, P.; Schelter, S.; Taptunov, A.; Lange, D.; Salinas, D. DataWig: Missing value imputation for tables. *J. Mach. Learn. Res.* **2019**, *20*, 1–6.
9. Aidos, H.; Tomás, P. Neighborhood-aware autoencoder for missing value imputation. In Proceedings of the 2020 28th European Signal Processing Conference (EUSIPCO), Amsterdam, NL, USA, 18–21 January 2021; pp. 1542–1546.
10. Ryu, S.; Kim, M.; Kim, H. Denoising autoencoder-based missing value imputation for smart meters. *IEEE Access* **2020**, *8*, 40656–40666. [\[CrossRef\]](#)
11. Gjørshoska, I.; Eftimov, T.; Trajanov, D. Missing value imputation in food composition data with denoising autoencoders. *J. Food Compos. Anal.* **2022**, *112*, 104638. [\[CrossRef\]](#)
12. Sánchez-Morales, A.; Sancho-Gómez, J.L.; Figueiras-Vidal, A.R. Complete autoencoders for classification with missing values. *Neural Comput. Appl.* **2021**, *33*, 1951–1957. [\[CrossRef\]](#)
13. Zhang, Y.; Zhou, B.; Cai, X.; Guo, W.; Ding, X.; Yuan, X. Missing value imputation in multivariate time series with end-to-end generative adversarial networks. *Inf. Sci.* **2021**, *551*, 67–82. [\[CrossRef\]](#)
14. Qiu, W.; Huang, Y.; Li, Q. IFGAN: Missing value imputation using feature-specific generative adversarial networks. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 4715–4723.
15. Zhao, F.; Lu, Y.; Li, X.; Wang, L.; Song, Y.; Fan, D.; Zhang, C.; Chen, X. Multiple imputation method of missing credit risk assessment data based on generative adversarial networks. *Appl. Soft Comput.* **2022**, *126*, 109273. [\[CrossRef\]](#)
16. Liu, Y.; Yu, R.; Zheng, S.; Zhan, E.; Yue, Y. Naomi: Non-autoregressive multiresolution sequence imputation. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
17. Kowsar, I.; Rabbani, S.B.; Samad, M.D. Attention-Based Imputation of Missing Values in Electronic Health Records Tabular Data. In Proceedings of the 2024 IEEE 12th International Conference on Healthcare Informatics (ICHI), Orlando, FL, USA, 3–6 June 2024; pp. 177–182.
18. Du, W.; Côté, D.; Liu, Y. Saits: Self-attention-based imputation for time series. *Expert Syst. Appl.* **2023**, *219*, 119619. [\[CrossRef\]](#)
19. Nie, T.; Qin, G.; Ma, W.; Mei, Y.; Sun, J. ImputeFormer: Low rankness-induced transformers for generalizable spatiotemporal imputation. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Barcelona, Spain, 25–29 August 2024; pp. 2260–2271.
20. Zhang, Z. Missing data imputation: Focusing on single imputation. *Ann. Transl. Med.* **2016**, *4*, 9. [\[PubMed\]](#)
21. Thomas, T.; Rajabi, E. A systematic review of machine learning-based missing value imputation techniques. *Data Technol. Appl.* **2021**, *55*, 558–585. [\[CrossRef\]](#)
22. Pujianto, U.; Wibawa, A.P.; Akbar, M.I. K-nearest neighbor (k-NN) based missing data imputation. In Proceedings of the 2019 5th International Conference on Science in Information Technology (ICSITech), Jonja, Indonesia, 23–24 October 2019; pp. 83–88.

23. Hasan, M.K.; Alam, M.A.; Roy, S.; Dutta, A.; Jawad, M.T.; Das, S. Missing value imputation affects the performance of machine learning: A review and analysis of the literature (2010–2021). *Inform. Med. Unlocked* **2021**, *27*, 100799. [\[CrossRef\]](#)
24. Read, J.; Pfahringer, B.; Holmes, G.; Frank, E. Classifier chains for multi-label classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 254–269.
25. Van Buuren, S.; Groothuis-Oudshoorn, K. mice: Multivariate imputation by chained equations in R. *J. Stat. Softw.* **2011**, *45*, 1–67. [\[CrossRef\]](#)
26. Tsai, C.F.; Hu, Y.H. Empirical comparison of supervised learning techniques for missing value imputation. *Knowl. Inf. Syst.* **2022**, *64*, 1047–1075. [\[CrossRef\]](#)
27. Yadav, M.L.; Roychoudhury, B. Handling missing values: A study of popular imputation packages in R. *Knowl.-Based Syst.* **2018**, *160*, 104–118. [\[CrossRef\]](#)
28. Wang, M.C.; Tsai, C.F.; Lin, W.C. Towards missing electric power data imputation for energy management systems. *Expert Syst. Appl.* **2021**, *174*, 114743. [\[CrossRef\]](#)
29. Palanivayagam, A.; Damaševičius, R. Effective handling of missing values in datasets for classification using machine learning methods. *Information* **2023**, *14*, 92. [\[CrossRef\]](#)
30. Li, L.; Wang, L.; Lu, L.; Zhu, T. Machine learning prediction of postoperative unplanned 30-day hospital readmission in older adult. *Front. Mol. Biosci.* **2022**, *9*, 910688. [\[CrossRef\]](#)
31. Raja, P.; Thangavel, K. Missing value imputation using unsupervised machine learning techniques. *Soft Comput.* **2020**, *24*, 4361–4392. [\[CrossRef\]](#)
32. Dubey, A.; Rasool, A. Efficient technique of microarray missing data imputation using clustering and weighted nearest neighbour. *Sci. Rep.* **2021**, *11*, 24297. [\[CrossRef\]](#)
33. Ribeiro, C.; Freitas, A.A. A data-driven missing value imputation approach for longitudinal datasets. *Artif. Intell. Rev.* **2021**, *54*, 6277–6307. [\[CrossRef\]](#)
34. Lai, X.; Zhang, L.; Liu, X. Takagi-sugeno modeling of incomplete data for missing value imputation with the use of alternate learning. *IEEE Access* **2020**, *8*, 83633–83644. [\[CrossRef\]](#)
35. Chen, X. An improved self-representation approach for missing value imputation. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 1450–1455.
36. Halder, B.; Ahmed, M.M.; Amagasa, T.; Isa, N.A.M.; Faisal, R.H.; Rahman, M.M. Missing information in imbalanced data stream: Fuzzy adaptive imputation approach. *Appl. Intell.* **2022**, *52*, 5561–5583. [\[CrossRef\]](#)
37. Han, J.; Kang, S. Active learning with missing values considering imputation uncertainty. *Knowl.-Based Syst.* **2021**, *224*, 107079. [\[CrossRef\]](#)
38. Cheng, C.H.; Chan, C.P.; Sheu, Y.J. A novel purity-based k nearest neighbors imputation method and its application in financial distress prediction. *Eng. Appl. Artif. Intell.* **2019**, *81*, 283–299. [\[CrossRef\]](#)
39. Almeida, A.; Brás, S.; Sargento, S.; Pinto, F.C. Focalize K-NN: An imputation algorithm for time series datasets. *Pattern Anal. Appl.* **2024**, *27*, 39. [\[CrossRef\]](#)
40. Duan, Y.; Lv, Y.; Kang, W.; Zhao, Y. A deep learning based approach for traffic data imputation. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 912–917.
41. Lin, W.C.; Tsai, C.F.; Zhong, J.R. Deep learning for missing value imputation of continuous data and the effect of data discretization. *Knowl.-Based Syst.* **2022**, *239*, 108079. [\[CrossRef\]](#)
42. Morales-Alvarez, P.; Gong, W.; Lamb, A.; Woodhead, S.; Peyton Jones, S.; Pawlowski, N.; Allamanis, M.; Zhang, C. Simultaneous missing value imputation and structure learning with groups. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 20011–20024.
43. Lee, M.; An, J.; Lee, Y. Missing-value imputation of continuous missing based on deep imputation network using correlations among multiple iot data streams in a smart space. *IEICE Trans. Inf. Syst.* **2019**, *102*, 289–298. [\[CrossRef\]](#)
44. Han, J.; Kang, S. Dynamic imputation for improved training of neural network with missing values. *Expert Syst. Appl.* **2022**, *194*, 116508. [\[CrossRef\]](#)
45. Liu, N.; Li, Y.; Zang, Z.; Hu, Y.; Fang, X.; Lolli, S. A deep learning-based imputation method for missing gaps in satellite aerosol products by fusing numerical model data. *Atmos. Environ.* **2024**, *325*, 120440. [\[CrossRef\]](#)
46. Cao, W.; Wang, D.; Li, J.; Zhou, H.; Li, L.; Li, Y. Brits: Bidirectional recurrent imputation for time series. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montreal, QC, Canada, 2–8 December; Volume 31.
47. Choudhury, S.J.; Pal, N.R. Imputation of missing data with neural networks for classification. *Knowl.-Based Syst.* **2019**, *182*, 104838. [\[CrossRef\]](#)
48. Psychogyios, K.; Ilias, L.; Ntanos, C.; Askounis, D. Missing value imputation methods for electronic health records. *IEEE Access* **2023**, *11*, 21562–21574. [\[CrossRef\]](#)
49. Yoon, J.; Jordon, J.; Schaar, M. Gain: Missing data imputation using generative adversarial nets. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 5689–5698.

50. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
51. Chen, Y.; Deng, W.; Fang, S.; Li, F.; Yang, N.T.; Zhang, Y.; Rasul, K.; Zhe, S.; Schneider, A.; Nevmyvaka, Y. Provably convergent schrödinger bridge with applications to probabilistic time series imputation. In Proceedings of the International Conference on Machine Learning, PMLR, Edmonton, AB, Canada, 23–29 July 2023; pp. 4485–4513.
52. Wang, X.; Zhang, H.; Wang, P.; Zhang, Y.; Wang, B.; Zhou, Z.; Wang, Y. An observed value consistent diffusion model for imputing missing values in multivariate time series. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, 6–10 August 2023; pp. 2409–2418.
53. Biloš, M.; Rasul, K.; Schneider, A.; Nevmyvaka, Y.; Günnemann, S. Modeling temporal data as continuous functions with stochastic process diffusion. In Proceedings of the International Conference on Machine Learning, PMLR, Honolulu, HI, USA, 23–29 July 2023; pp. 2452–2470.
54. Dai, Z.; Getzen, E.; Long, Q. SADI: Similarity-Aware Diffusion Model-Based Imputation for Incomplete Temporal EHR Data. In Proceedings of the International Conference on Artificial Intelligence and Statistics, PMLR, Valencia, Spain, 2–4 May 2024; pp. 4195–4203.
55. Yang, X.; Sun, Y.; Chen, X. Frequency-aware generative models for multivariate time series imputation. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 52595–52623.
56. Zhou, J.; Li, J.; Zheng, G.; Wang, X.; Zhou, C. Mtsci: A conditional diffusion model for multivariate time series consistent imputation. In Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, Boise, IO, USA, 21–25 October 2024; pp. 3474–3483.
57. Aljrees, T. Improving prediction of cervical cancer using KNN imputer and multi-model ensemble learning. *PLoS ONE* **2024**, *19*, e0295632. [[CrossRef](#)]
58. Shahjaman, M.; Rahman, M.R.; Islam, T.; Auwul, M.R.; Moni, M.A.; Mollah, M.N.H. rMisbeta: A robust missing value imputation approach in transcriptomics and metabolomics data. *Comput. Biol. Med.* **2021**, *138*, 104911. [[CrossRef](#)]
59. Ae Lee, J.; Gill, J. Missing value imputation for physical activity data measured by accelerometer. *Stat. Methods Med. Res.* **2018**, *27*, 490–506. [[CrossRef](#)]
60. Huan, J.; Li, M.; Xu, X.; Zhang, H.; Yang, B.; Jianming, J.; Shi, B. Multi-step prediction of dissolved oxygen in rivers based on random forest missing value imputation and attention mechanism coupled with recurrent neural network. *Water Supply* **2022**, *22*, 5480–5493. [[CrossRef](#)]
61. Zhu, X.; Wang, J.; Sun, B.; Ren, C.; Yang, T.; Ding, J. An efficient ensemble method for missing value imputation in microarray gene expression data. *BMC Bioinform.* **2021**, *22*, 188. [[CrossRef](#)] [[PubMed](#)]
62. Rao, R.S.; Kalabarige, L.R.; Alankar, B.; Sahu, A.K. Multimodal imputation-based stacked ensemble for prediction and classification of air quality index in Indian cities. *Comput. Electr. Eng.* **2024**, *114*, 109098. [[CrossRef](#)]
63. Jung, S.; Moon, J.; Park, S.; Rho, S.; Baik, S.W.; Hwang, E. Bagging ensemble of multilayer perceptrons for missing electricity consumption data imputation. *Sensors* **2020**, *20*, 1772. [[CrossRef](#)]
64. Samad, M.D.; Abrar, S.; Diawara, N. Missing value estimation using clustering and deep learning within multiple imputation framework. *Knowl.-Based Syst.* **2022**, *249*, 108968. [[CrossRef](#)] [[PubMed](#)]
65. Wang, T.; Ke, H.; Jolfaei, A.; Wen, S.; Haghighi, M.S.; Huang, S. Missing value filling based on the collaboration of cloud and edge in artificial intelligence of things. *IEEE Trans. Ind. Inform.* **2021**, *18*, 5394–5402. [[CrossRef](#)]
66. Jiang, D.; Zhang, S. An explainable missing data imputation method and its application in soft sensing. *Measurement* **2025**, *2025*, 117692. [[CrossRef](#)]
67. Peng, D.; Zou, M.; Liu, C.; Lu, J. RESI: A region-splitting imputation method for different types of missing data. *Expert Syst. Appl.* **2021**, *168*, 114425. [[CrossRef](#)]
68. Khan, S.I.; Hoque, A.S.M.L. SICE: An improved missing data imputation technique. *J. Big Data* **2020**, *7*, 37. [[CrossRef](#)]
69. Matzka, S. Explainable artificial intelligence for predictive maintenance applications. In Proceedings of the 2020 Third International Conference on Artificial Intelligence for Industries (AI4I), Irvine, CA, USA, 21–23 September 2020; pp. 69–74.
70. Moro, S.; Cortez, P.; Rita, P. A data-driven approach to predict the success of bank telemarketing. *Decis. Support Syst.* **2014**, *62*, 22–31. [[CrossRef](#)]
71. Yeh, I.C.; Yang, K.J.; Ting, T.M. Knowledge discovery on RFM model using Bernoulli sequence. *Expert Syst. Appl.* **2009**, *36*, 5866–5871. [[CrossRef](#)]
72. Rahman, M.A.; Islam, M.Z.; Bossomaier, T. ModEx and Seed-Detective: Two novel techniques for high quality clustering by using good initial seeds in K-Means. *J. King Saud-Univ.-Comput. Inf. Sci.* **2015**, *27*, 113–128. [[CrossRef](#)]
73. Kalina, J. High-dimensional data in economics and their (robust) analysis. *Serbian J. Manag.* **2017**, *12*, 157–169. [[CrossRef](#)]
74. Salzberg, S.L. *Exemplar-Based Learning: Theory and Implementation*; Aiken Computation Laboratory, Center for Research in Computing Technology, Harvard University: Cambridge, MA, USA, 1988.

75. Egli, D.B.; Bruening, W.P. Potential of early-maturing soybean cultivars in late plantings. *Agron. J.* **2000**, *92*, 532–537. [\[CrossRef\]](#)
76. Landwehr, J.M.; Pregibon, D.; Shoemaker, A.C. Graphical methods for assessing logistic regression models. *J. Am. Stat. Assoc.* **1984**, *79*, 61–71. [\[CrossRef\]](#)
77. Rosly, R.; Makhtar, M.; Awang, M.K.; Awang, M.I.; Rahman, M.N.A. Analyzing performance of classifiers for medical datasets. *Int. J. Eng. Technol. (UAE)* **2018**, *7*, 136–138. [\[CrossRef\]](#)
78. Unwin, A.; Kleinman, K. The iris data set: In search of the source of virginica. *Significance* **2021**, *18*, 26–29. [\[CrossRef\]](#)
79. Phiwhorm, K.; Saikaew, C.; Leung, C.K.; Polpinit, P.; Saikaew, K.R. Adaptive multiple imputations of missing values using the class center. *J. Big Data* **2022**, *9*, 52. [\[CrossRef\]](#)
80. Ferenc, D.; MAGIC Collaboration. The MAGIC gamma-ray observatory. *Nucl. Instruments Methods Phys. Res. Sect. A Accel. Spectrometers Detect. Assoc. Equip.* **2005**, *553*, 274–281. [\[CrossRef\]](#)
81. Ahmed, M.; Kashem, M.A.; Rahman, M.; Khatun, S. Review and analysis of risk factor of maternal health in remote area using the Internet of Things (IoT). In Proceedings of the InECCE2019—5th International Conference on Electrical, Control & Computer Engineering, Kuantan, Malaysia, 29 July 2019; Springer: Berlin/Heidelberg, Germany, 2020; pp. 357–365.
82. Thrun, S.; Bala, J.; Bloedorn, E.; Bratko, I.; Cestnik, B.; Cheng, J.; Keller, S.; Kononenko, I.; Kreuziger, J.; Michalski, R.; et al. *The Monk's Problems: A Performance Comparison of Different Learning Algorithms*; cmu-cs-91-197; Carnegie Mellon University: Pittsburgh, PA, USA, 1991.
83. Dinh, A.; Miertschin, S.; Young, A.; Mohanty, S.D. A data-driven approach to predicting diabetes and cardiovascular disease with machine learning. *BMC Med. Inform. Decis. Mak.* **2019**, *19*, 211. [\[CrossRef\]](#)
84. Palechor, F.M.; de la Hoz Manotas, A. Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico. *Data Brief* **2019**, *25*, 104344. [\[CrossRef\]](#)
85. Cinar, I.; Koklu, M. Classification of rice varieties using artificial intelligence methods. *Int. J. Intell. Syst. Appl. Eng.* **2019**, *7*, 188–194. [\[CrossRef\]](#)
86. Lakshmi, B.J.; Madhuri, K.; Shashi, M. An efficient algorithm for density based subspace clustering with dynamic parameter setting. *Int. J. Inf. Technol. Comput. Sci.* **2017**, *9*, 27–33. [\[CrossRef\]](#)
87. Hofmeyr, D.P. Degrees of freedom and model selection for k-means clustering. *Comput. Stat. Data Anal.* **2020**, *149*, 106974. [\[CrossRef\]](#)
88. Chai, T.; Draxler, R.R. Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature. *Geosci. Model Dev.* **2014**, *7*, 1247–1250. [\[CrossRef\]](#)
89. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
90. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.