Original Research Article

Enhancing Missing Data Imputation with Improved DAE Training and Input Recombination

ABSTRACT

The increasing significance of addressing data loss, has led to a heightened focus on missing data imputation (MDI). Autoencoder (AE) models, renowned for their ability to autonomously learn and impute missing data, are gaining prominence in MDI. These models exhibit adaptability to diverse datasets, and their unsupervised nature makes them robust in handling data lacking clear labels.

This study aims to explore the scope and objectives ofAE training, which encompass critical elements such as optimization algorithms, loss functions, and training epochs. We specifically investigate the impact of updating input data during AE training, a topic that has been insufficiently explored in existing research.Traditionally, AEs are trained on the original data, assuming it contains latent information. However, in the context of MDI, where data may be corrupted, it becomes imperative to evaluate whether updating input data can lead to superior results.

The objective of this research is to introduce and evaluate two methods inspired by Gradient Boosting Machines: Short-Term Reconstruction with Iterative Updates (STR-IU) and Long-Term Reconstruction with a Single Update (LTR-SU).We utilize DenoisingAutoencoder (DAE) models and examine how various optimization mechanisms affect our proposed methods. We conduct comparisons between Stochastic Gradient Descent (SGD) and the Adam optimization algorithm, and transform three distinct datasets into synthetic datasets with varying levels of missing data (5%, 15%, 25%). The results indicate that, while performance may not consistently excel across all training epoch settings, there is a noticeable overall improvement when updating input data, whether using SGD or Adam. Additionally, LTR-SU outperforms STR-IU, and models with DAE using SGD exhibit greater optimization compared to those using Adam

Keywords: Missing data imputation; DenoisingAutoencoder; updating input data

1. INTRODUCTION

There are numerous reasons that can lead to missing data, such as equipment malfunction, human error, limitations in data collection (e.g.,patients or clients dropping out or missing appointments), and transmission loss (e.g., an unstable network signal). Missing data can have a detrimental effect on data quality and can cause issues during data analysis [1-2]. Without proper handling of missing data, it can increase computational costs, skew outcomes, and misguide researchers [3].

Missing data can be problematic, especially when there is a large amount of missing data. This can make it challenging to identify statistical effects because the statistical power of the data is diminished[40]. Additionally, missing data can introduce bias to the parameter estimates being studied because the available observational data may not accurately represent the population. Decreased precision of an estimate means that it will be less accurate [4-5]. As such, missing

data should be handled with care. If missing data is not properly imputed, it can lead to increased uncertainty in the analysis and incomplete conclusions.

Based on the relationship between the missing values and other observed variables in the dataset, missing data can be categorized into three mechanisms of missingness [6-8].

1.1. CATEGORIZATION OF MISSINGNESS MECHANISMS

1.1.1Missing Completely At Random (MCAR)

MCAR denotes that the missing values in the dataset are completely random and unrelated to the values of other observed variables. In other words, the probability of a value being missing is not related to the values of other variables in the dataset.

1.1.2 Missing At Random (MAR)

In contrast, MAR signifies that the missingness is related to the observed variables in the dataset. In other words, the probability of a value being missing is dependent on the values of other observed variables, but not on the other missing values.

1.1.3 Missing Not At Random (MNAR)

MNAR indicates that the missingness is related to the missing value itself or with other unobserved data. In other words, the probability of a value being missing depends on the values of other unobserved variables or the missing value itself.

1.2. MISSING DATA IMPUTATION IN MCAR, MAR AND MNAR

The approaches to handling missing data may depend on the missing mechanisms and the proportions of missing data [9]. It is commonly recommended that when dealing with MCAR or MAR mechanisms, if the proportion of missing data is less than 5%, it can be suggested to omit or use traditional techniques (e.g., deletion or single imputation) to handle it [10-12]. This is because the influence of bias is likely to be minimal. However, if the proportion of missing data is over 5%, more advanced techniques (e.g., multiple imputation, model-based procedures, and machine learning methods) should be used to fill in the missing data [9].

On the other hand, due to the missingness mechanism of MNAR being dependent on the missing value itself or other unobserved data, researchers may use model-based methods or shared parameter models, which may require strong assumptions to more explicitly resolve the missingness mechanism. In order to account for missingness due to selection bias, sensitivity analysis and correction factors may be used to evaluate the robustness of the results to different assumptions [13].

1.3. APPLICATION OF AUTOENCODERS FOR MISSING DATAIMPUTATION

In recent years, deep learning-based methods have been successfully applied to many research problems, including the missing data imputation community. Among all deep learning methods, the Autoencoder (AE) is particularly noteworthy. Firstly, AE is an ANN (Artificial Neural Networks)-based model, which can be trained in an unsupervised way and is designed to learn from incomplete data and produce new probable values for imputation. With the benefits of ANNs, AEs can model much more complex data patterns compared to other simpler methods (e.g., the KNN method based on simple distances or the MICE method, which relies internally on simple regressions).

Secondly, since AEs can learn directly from incomplete data, they natively support multivariate scenarios. This means that AE can perform the imputation of all features with missing values while accounting for the relations between them, which is more difficult for most imputation methods and vanilla ANNs. A comprehensive study[14] surveyed the use of AEs for the imputation of tabular data and considered 26 works published between 2014 and 2020. The analysis results showed that both the variants of AE (Denoising AE and Variational AE) outperformed their competitors in the vast majority of works that reported imputation results.

The use of AEs for missing data imputation is a promising area of researchandVarious types of AEs have been investigated for this task, including Denoising AE [15-18], Variational AE [19-21], as well as other variants of AEs [22-26].

However, it is important to note that using AEs for missing data imputation is still in its early stages, and there is ongoing research to explore their effectiveness and limitations.

One potential limitation of AEs is that they directly learn from input data, making them more susceptible to noise and outliers. Additionally, the fitting capacity of AEs is limited, which can lead to overfitting and a weakening of the model's generalization ability. As a result, it is crucial to carefully consider the use of AEs for missing data imputation and explore ways to mitigate these limitations. Researchers have proposed various modifications to AEs, such as adding regularization terms to the loss function, using different activation functions or network architectures, and incorporating external information or constraints into the model.

1.4. THE PROPOSED INTEGRATION SCHEME

We propose an innovative approach involving the integration of simulated data, generated through the training of an AE for imputation, with the incomplete original dataset. This strategic integration aims to enhance training performance by supplementing deficient information within the input data using the imputed values derived from the AE training process. The amalgamation of simulated and original data is anticipated to provide a more comprehensive and accurate representation during subsequent training iterations, contributing to improved imputation outcomes.

Overall, the integration of simulated data with the original dataset addresses the need for enhanced input data quality, particularly in the context of missing data imputation. This strategic integration leverages the capabilities of AE to learn from incomplete data and generate probable values, thereby offering a promising approach to mitigate the limitations associated with incomplete datasets. Through this integration, the training process is enriched with more informative data, leading to improved imputation outcomes and better model performance.

---

2. RELATED WORK

2.1 APPROACHES TO MISSING DATA IMPUTATION

handling missing data can be broadly divided into two categories: statistical-based methods (e.g., single imputation, multiple imputation, and model-based procedures), which aim to restore the missing values with the most similar ones among the observed data, and machine learning-based techniques which replace the missing values by constructing a predictive model with the available data to estimate values.[14]

2.1.1 Statistical Methods for Missing Data Imputation

The most common traditional statistical-based methods used to handle missing datainclude deletion and single imputation

2.1.1.1 Deletion Methods

Deletion techniques are intuitive and easy to execute, aiming to solve the problem by deleting the cases that contain missing data. Based on the type of deletion case analysis, there are two techniques:Complete-case analysis and available case analysis. Complete-case analysis involves removing all data from an analysis that contains missing values. However, this technique introduces significant bias if there are a large number of missing values or if the original data set is too small.On the other hand available-case analysis is a more selective method, which decides the quantity of missing data on a case-by-case basis, deleting cases with high levels of missing data to minimize data loss. Both complete-case analysis and available-case analysis assume the MCAR mechanism.

2.1.1.2 Single Imputation Methods

Single imputation is a procedure that involves analyzing the data together with other variables to find the most likely value that can substitute for the missingdata. There are several methods to find the imputation value, such as by themean, median, ormode of the available values of that variable.

• Mean/Median Imputation:
This technique involves replacing the missing value with the arithmetic meanormedian of all the other cases, This method is suitable for small data samples because results can be distorted due to uncertainty in the sample distribution.

• Regression Imputation:

Regression imputation uses a function to estimate the relationship between a data point and its related variable. Commonly, the least-squares fit form is used, but other forms such as multiple linear, quadratic, cubic, and non-polynomial models are also employed. A limitation of regression imputation is thatthe chosen regression model may fit perfectly but lack an error term to account for data variability

• Hot Deck and Cold DeckImputation:

The hot-deck technique involves finding a similar matched dataset, using approaches like the distance function approach, which imputes the missing value by the smallest squared distance of the case with the missing value, or the pattern matching approach, grouping similar cases together and randomly selectingthe imputed value from a similar group. In contrast,the cold-deck techniqueis similar to hot-deck imputation, but relies on external sources of information, not derived from the current dataset.

2.1.1.3MultipleImputationMethods

To address the limitations of single imputation, which may inadequately capture the variability in datasets and potentially introduce bias, researchers propose multiple imputation techniques. The multiple imputation procedure involves analyzing incomplete datasets through various simulation models. introducing variability to theimputed data, andgenerating a spectrum of plausible values for each missing data point.

In the initial phase, missing values in incomplete datasets are imputed over multiple iterations. During each iteration, one or more sets of imputations are generated using single imputation techniques, resulting in a comprehensive set of imputations for all missing values of the target variable. Standard errors are computed for each iteration. Upon completion of the m iterations, the estimated values and standard errors are integrated to produce a final result, typically represented as an averaged set of values.

While multiple imputation mitigates the bias associated with single imputation by incorporating dataset variability and providing a range of plausible responses, its implementation is intricateand time-consuming due to the need to run multiple iterations for estimation purposes. Additionally, careful design is required for combining results and ensuring the correct utilization of estimated values, underscoring the importance of methodological rigor in the application of multiple imputation techniques.

2.1.2 Machine Learning-based Methods for Missing Data Imputation

Machine learning methods involve creatingand constructing algorithms that make predictions on missing databased on theinformation available in the dataset. Some ofthe most common machine learning techniques are mentioned below

2.1.2.1 K-Nearest Neighbors (KNN) Algorithm

TheKNN algorithm is widely employed for imputing missing values by replacing them with observed values from the nearest neighbors. For each observation with missing values, the algorithm identifies the nearest neighbors based on a predefined distance function,quantifying the similarity or dissimilarity between observations. Common metrics include Euclidean distance, Manhattan distance, or other suitable measures depending on the characteristics of the data. The imputation is performed by incorporating observed values from the nearest neighbors, leveraging the local data structure to make informed and contextually relevant replacements.

2.1.2.2Decision Trees (DT)

Decision trees are widely used tools in supervised learning, represented as tree structures and employed to map data and observations. The primary objective of decision trees is to draw conclusions about the target values, which are represented in the tree's leaf nodes. One major advantage lies in the ability to visualize data, allowing for easy comprehension of the data structure. The goal is to find the optimal decision tree by minimizing the standard error, making decision trees an interpretable model that provides a clear illustration of data mapping and prediction processes

2.1.2.3Neural Networks (NN)

A neural network is comprised of layers of artificial neurons, collectively processing and transforming input data into meaningful output. Each connection between neurons is assigned a weight, and the network refines its performance by adjusting these weights during the training process.

Through iterative adjustments during training, the neural network minimizes the disparity between predicted and actual outcomes. , enabling generalization to unseen data.

Neural networks are particularly effective in capturing non-linear relationships within data, making them powerful tools for tasks like pattern recognition, classification, and regression. Their proficiency in handling complex and high-dimensional datasets allows them to learn and represent intricate patterns and dependencies.

2.2 AUTOENCODER (AE)

An AE is a type of artificial neural network composed of at least three layers: an input layer, a hidden layer, and an output layer. A typical AE consists of two components: an encoder and a decoder. In the encoder part, the input vector X is mapped to a latent feature Z through involves nonlinear encoding functions. In the decoder part, the latent feature Z is mapped back to reconstruct the original input X through nonlinear decoding functions. In summary, an AE utilizes nonlinear encoding and decoding functions to map an input to a latent representation and then reconstruct the input from that latent representation.

The encoder mapping process is a nonlinear transformation represented as f(x) = σ(WX_e + b_e), where σ represents a nonlinearity such as the activation function, W represents the weight matrix, and b represents the bias vector. Similarly, in the decoder part, the latent space Z is mapped back to reconstruct the input as the output Y, represented as g(z) = σ(W_dZ + b_d), where σ is again a nonlinearity, W' represents the transpose of the weight matrix used in the encoder, and b' represents the bias vector. These functions f(x) and g(z) capture the nonlinear relationships between the input and the latent representation, as well as between the latent representation and the reconstructed output. The structure of the Villa AE is illustrated in Figure 1.

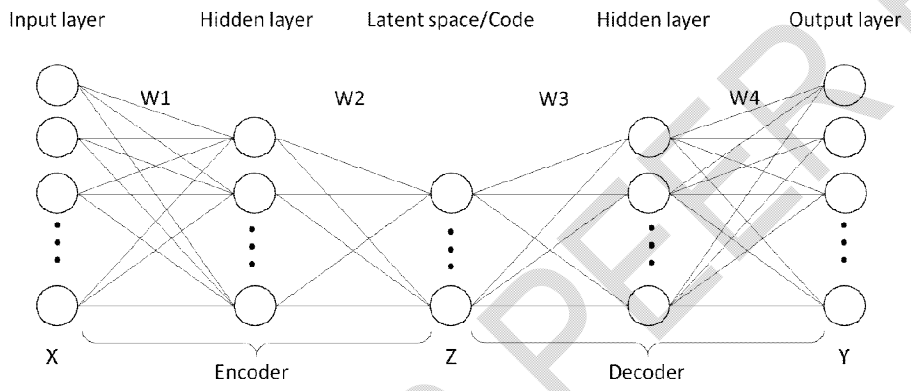


Figure 1. Structure of the vanilla AE.

The goal of an AE is to minimize the reconstruction error between the input X and the output Y. The training process involves optimizing the model parameters $\{W_e, W_d, b_e, b_d\}$) to achieve this objective. To optimize these parameters, a Stochastic Gradient Descent (SGD) variant is commonly used. SGD is an iterative optimization algorithm that updates the parameters based on the gradients of the loss function with respect to the parameters. The specific variant of SGD used can vary, and popular choices include Adam, RMSprop, or plain vanilla SGD. During training, the AE calculates the reconstruction error, which is the discrepancy between the input X and the output Y. This error is then used to compute a loss function, quantifying the overall reconstruction performance. The choice of loss function depends on the nature of the data and the specific task at hand. Mean squared error (MSE) is a common choice for continuous data, while binary cross-entropy or categorical cross-entropy may be used for binary or categorical data, respectively. The AE iteratively adjusts the model parameters using the SGD variant, aiming to minimize the reconstruction error and improve the overall performance of the AE. The optimization process continues until the model converges or reaches a predefined stopping criterion.

2.2.1 Denoising Autoencoder (DAE) Application in Missing Data Imputation

The DAE[15] is a variant of AE specifically designed to handle noisy data, such as data corruption or missing data. Its denoising property makes it particularly useful for missing data imputation. Instead of using the original input X, theDAE corrupts the input vector X to obtain a partially corrupted input X̃. The encoding function of DAE is defined as f(X̃) = σ(WX̃ + b), where σ represents the activation function. In practical applications, the replacement of the input X can be achieved by adding additional noise according to a certain distribution or setting specific elements of the input to zero. Alternatively,

different corruption functions can be utilized to substitute selected data with prepared values that represent the missing status.

In [16],a study, a Daily Load Profile (DLP) based missing value imputation framework was proposed, utilizing a DAE. The framework consists of four stages. In Stage 1, the ground-truth data is constructed from the dataset by replacing missing values with the average of previous time slots, ensuring the availability of complete data for training and evaluation. Stage 2 involves implementing data corruption using two strategies: random corruption and block-wise missing, introducing artificial missing values into the dataset, to simulate real-world missing data. In Stage 3, the AE structure is selected, including the number of layers, neurons, and activation functions, and the DAE network is built based on these choices. Additionally, three different loss function configurations are explored to optimize the imputation performance. Finally, in Stage 4, the imputation performance is compared across six different configurationsbased on three corruption functions and two test set formulations, assessing the effectiveness of the DAE framework in imputing missing values in the DLP dataset.

To address the sensitivity to initial imputation in the DAE,.A technique called metamorphic truth[17] is employed. This technique involves modifying the truth reported to the optimizer based on the model's predictions, allowing for a more flexible learning process.Additionally, the predicted values are usedto impute subsequent training of the DAE, reducing the discrepancy between the input and output and improving the robustness and effectiveness of the DAE for imputation tasks.

In the context of imputing missing data in multivariate time series, a bidirectional Long Short-Term Memory (LSTM)[18] blockis employed as an encoder. This LSTM block takes the entire series as input and generates a representation for each time step, effectively capturing temporal information. Similar to the DAE approach, the input is corruptedby randomly setting certain values to zero, and the model is trained to reconstruct the clean input for each time step, learning the correlation between variables and effectively imputing missing values in the multivariate time series.

2.2.2. Variational Autoencoder (VAE) Application in Missing Data Imputation

The VAE extends the capabilities of AE by incorporating probabilistic modeling and inference through variational Bayesian methods.

Indeed, the VAE is a generative model that maps the input space X to the latent space Z. Unlike the vanilla AE, which focuses on minimizing the error between the input X and the reconstructed output Y, the VAE takes a probabilistic approach, aiming to maximize the probability distribution of the data p_θ(X) in terms of the model parameters θ, which is equivalent to solving a maximum likelihood problem.

In Equation (1), the marginal likelihood p_θ(X) is defined as the integral of the joint probability distribution p_θ(X,Z)over the latent variable Z. However, computing this integral directly is often intractable because both the latent variable Z and the generative model parameter θ are unknown.

$$p_\theta(X) = \int p_\theta(X,Z)dz = \int p_\theta(Z)p_\theta(X|Z)dz \quad (1)$$

To overcome this problem, [19-20] proposed training the generative model jointly with an inference model using variational inference. The data model consisting of the generative model p_θ(X,Z) = p_θ(Z) p_θ(X|Z) and the inference model q_θ(Z|X) parameterized with ∅ which approximates the true posterior p_θ(X|Z).

By optimizing the terms of ∅ and θ simultaneously, a variational lower bound on the marginal likelihood can be derived. This is achieved by maximizing the objective function L(θ,∅,X) defined in Equation (2). The objective function consists of two terms: the expected reconstruction error, which measures the similarity between the input X and the reconstructed output given the approximate posterior distribution, and the Kullback-Leibler (KL) divergence between the approximate posterior and the prior p_θ(X).

$$L(\theta,\emptyset,X) = E_{q\emptyset(Z|X)}[\log p_\theta(X|Z)] - KL[q_\theta(Z|X)||p_\theta(Z)] \quad (2)$$

The KL divergence acts as a regularization term, ensuring that the learned latent distribution matches a predefined prior distribution, such as a multivariate Gaussian. This encourages similar input data to be represented by similar latent spaces.

In summary, the VAE learns a generative model that maps the input space X to the latent space Z. By incorporating an inference model and using variational inference, the VAE approximates the true posterior distribution and optimizes the model parameters to maximize the likelihood of the data. The objective function of the VAE includes a reconstruction term and a regularization term to balance the reconstruction accuracy and adherence to the prior distribution.

However, empirical studies have highlighted that VAEs may result in underestimation and overconfident imputationswhen handling missing data. To address this limitation, the application of β-VAE[21]has been proposed, offering a framework for approximate Bayesian inference of deep generative models using the power likelihood,Whichhas demonstrated robustness against model misspecification. Determining an appropriate value of β is crucial, and cross-validation is employedto fine-tune βfor accurate multiple imputation..

In [22], the authors introduce two innovative submodels based on Deep VAE (DVAE). These submodels serve as the foundation for developing two distinct soft sensor frameworks designed to handle scenarios with and without missing data. The first submodel, called Supervised DVAE (SDVAE) incorporates both samples and labels, enhancing prediction accuracy. The second submodel, known as Modified Unsupervised DVAE (MUDVAE), modifies the benchmark latent distribution to align with that learned by SDVAE. This modification is motivated by the observation that when the KL divergence between the learned latent distribution of MUDVAE and the modified benchmark latent distribution is sufficiently small, sampling from MUDVAE's latent distribution is equivalent to sampling from SDVAE's, enabling predictions for test samples.

Adversarial AE (AAE)[23] incorporate adversarial training. AAEs build upon the framework of VAEs but utilize a different loss computation, specifically the adversarial loss commonly used in Generative Adversarial Network (GAN). AAEs consist of three main components: the encoder, decoder, and discriminator networks. The encoder and decoder are similar to those in VAEs, and utilizing a discriminator network to distinguish between samples drawn from the latent space and those sampled from the prior distribution, improving the quality of generated samples

Similarly,, Wasserstein AEs (WAE)[24]employ the Wasserstein distance as a metric for training,optimizing the learned latent space distribution to facilitate more accurate and meaningful data generation.

2.2.3 Other Variants of Autoencoders in Missing Data Imputation

Beyond DAEs and VAEs, studies have focused on the structural aspects of AEs to enhanceperformance and feature extraction capabilities.

.

Stacked AE (SAE)[25], also known as the Deep AE, which incorporates multiple layers of encoder and decoder units. , learning increasingly abstract representations through a hierarchical structure.The training process of the SAE starts from the bottom layer and proceeds in a layer-wise manner. Each layer is pretrained individually as an AE, where it learns to reconstruct its input data. This pretraining step initializes the weights of the network and helps in capturing meaningful initial representations. After pretraining, the entire network is fine-tuned jointly using backpropagation and gradient-based optimization algorithms, allowing for the refinement of the learned representations.

Sparse AE (SAE)[26], which incorporatessparsity constraints during training, learning more efficient and concise representations of input data. The goal of the SAE is to enforce that only a small fraction of neurons or hidden units in the AE are active for each input sample. This sparsity property allows the SAE to learn more efficient and concise representations of the input data. One common method to achieve sparsity is by adding a sparsity regularization term to the loss function, which encourages a limited number of hidden units to activate for each input, promoting sparsity in the learned representations.

Combining sparse representations with hierarchical learning allows for the extraction of increasingly abstract features while promoting compact representations..

A method [27]leveraging both row-row and column-column relationships collaboratively uses two AEs: a user-based AE capturing column-column relationships and an item-based AE capturing row-row relationships. The final imputations, are obtained by averagingthe predictions from both AEs, improving the accuracy and reliability of the imputed values.

A missing data imputation optimization method called SAE-CD[28], combines the use of a Sparse AE (SAE) model with a Coordinate Descent (CD) algorithm. First, an SAE model is trained to learn a compressed representation of the input data while capturing its essential features. By minimizing the reconstruction error, the SAE model can effectively fill in missing values and impute the incomplete data. Subsequently, the CD algorithm is employed to refine the imputations. The CD

algorithm optimizes the imputed values by adjusting them in a coordinate-wise manner, ensuring the authenticity and accuracy of the imputations, iterativelyrefining imputations to enhance quality.

An ensemble modeling technique applied to DAEs [29], involves training multiple DAEs on different subsets of the data. , combining local transform functions to form the final transform function. To ensure the robustness and generalization of the ensemble model, a linear weighting function is applied to each training sample, the global mapping function is estimated as:

$$f(\cdot) \triangleq \lambda_1(\cdot) f_1(\cdot) + \lambda_2(\cdot) f_2(\cdot) \quad (3)$$

Where $\lambda_1(\cdot)$ and $\lambda_2(\cdot)$ are the weighting coefficient functions. And for overcoming overfitting problem, $0 \leq \lambda_1(\cdot)$, $\lambda_2(\cdot) \leq 1$ and $\lambda_1(\cdot) + \lambda_2(\cdot) = 1$ constraintsare added, to prevent overfitting, ensuring robust and accurate data representation.

An Orthogonal-Least-Squares (OLS)-based AE [30] incrementally addsneurons to the hidden layer using the OLS method, maximizing the increase in explained variance and promoting orthogonality among hidden neurons, Orthogonality promotes diversity among the learned representations, preventing redundancy and facilitating more effective feature extraction. The proposed OLS-based AE can helps address the overfitting issue that can arise when using a large number of hidden neurons. Additionally. It offers an approach to optimize the composition of the hidden layer, ensuring the extraction of relevant and non-redundant features, which can lead to improved performance and generalization capabilities in various applications.

The Monte Carlo Dropout (MCD) [31] technique within AE and VAE layers generates multiple imputed values for each missing data point, capturing the underlying distribution without relying solely on individual data points,and averaging the output values for comprehensive imputation.To obtain the final imputed value, the output values from the model are averaged. This averaging process provides a more comprehensive representation of the imputed value, incorporating the uncertainty and variability captured by the multiple generated inputs. The MCD-AE model improves the accuracy of imputation by considering a range of possible values for each missing data point, resulting in more robust and reliable imputation results,

For traffic data imputation,the Ensemble Convolutional AE[32]uses multiple AEs trained with different input feature mapsusing a parameterized data aggregation rule. These feature maps are created by filling the missing positions with zero values and historical average values, respectively.In the ensemble model, the AEs process the input feature maps, denoted as X^1 (with missing positions filled with zeros) and X^2 (with missing positions filled with historical average values). Consequently, two matricized outputs, Y^1 andY^2, are generated.

$$Y = \alpha \times Y^1 + (1 - \alpha) \times Y^2 \quad (4)$$

where α is a self-adapting weight that determines the contribution of each autoencoder's output, which are updated using backpropagation during the training processto achieve superior imputation accuracy and robustness.

2.2.4 Research Gaps and Recent Works

To highlight the research gaps and recent works, we present the following table summarizing the current gaps and recent advancements in the field. It is important to note that while various studies have been conducted in the field of missing data imputation (MDI), none specifically address the integration of simulated data with the original dataset as proposed in Section 1.4. Our proposed integration scheme, which involves the amalgamation of imputed values derived from AE training with the incomplete original dataset, stands out as an innovative approach in this domain.

Table1. Research Gaps and Recent Advances in Missing Data Imputation

Research Area	Existing Approaches	Research Gaps	Recent Works
Statistical Methods	-Mean/Median Imputation -Regression Imputation -Hot Deck/Cold Deck -Multiple Imputation	-Fails to capture underlying data variability -Bias in imputed values -Time-consuming implementation of multiple imputation	N/A
Machine Learning Methods	-K-Nearest Neighbors (KNN) -Decision Trees (DT) -Neural Networks (NN)	-Limited by local data structure (KNN) -Interpretability issues (NN) -Lack of comprehensive temporal handling (DT, NN)	N/A

Autoencoder (AE) Methods	- Vanilla AE -Denoising AE (DAE) -Variational AE (VAE)	-Limited robustness in noisy and incomplete data -Underestimation and overconfident imputations (VAE)	- Daily Load Profile (DLP) based DAE[16] -Metamorphic truth and bidirectional LSTM for DAE[17][18] -β-VAE for model misspecification[21]
Stacked AE (SAE)	-Hierarchical representation learning -Layer-wise training and fine-tuning	-Complexity and computational cost -Capturing sparse activations	-Layer-wise pretraining and joint fine-tuning of SAE[25]
Sparse AE (SAE)	- Incorporation of sparsity constraints during training	- Efficient and concise representation learning	- Sparse AE for compact data representation[26]
Double AE	- Leveraging row-row and column-column relationships collaboratively	- Balancing and combining predictions from both AEs	- Double AEs for improved imputation[27]
SAE-CD	- Combining Sparse AE with Coordinate Descent (CD) algorithm	- Iterative refinement of imputed values	- SAE-CD for optimizing imputation using SAE and CD[28]
Ensemble Modeling with DAEs	- Training multiple DAEs on different data subsets	- Ensuring robustness and generalization in ensemble model	- Ensemble DAEs with linear weighting function[29]
OLS-based AE	- Incremental addition of neurons using OLS method	- Addressing overfitting issues - Optimizing hidden layer composition	- OLS-based AE for orthogonal feature extraction and overfitting prevention[30]
MCD in AE/VAE	- Generating multiple imputed values through MCD technique	- Ensuring accuracy and reliability in imputed values	- MCD in AE/VAE layers for improved imputation[31]
Ensemble Convolutional AE	- Handling spatial-temporal features in traffic data	- Capturing complex spatial-temporal dependencies	- Ensemble Convolutional AE for traffic data imputation[32]

Despite the advancements in MDI, there is a notable lack of research specifically focusing on the integration of simulated data with the original dataset to enhance training performance, as proposed in our innovative approach in Section 1.4. This unique strategy leverages the capabilities of AE to learn from incomplete data and generate probable values, thereby offering a promising approach to mitigate the limitations associated with incomplete datasets. Our proposed method addresses the need for enhanced input data quality and provides a more comprehensive and accurate representation during subsequent training iterations, contributing to improved imputation outcomes and better model performance.

2.3. GRADIENT BOOSTING MACHINES (GBM)

Gradient Boosting Machines (GBM)[33] are a machine learning algorithm primarily used for supervised learning tasks, such as regression and classification. Falling under the category of boosting ensemble methods, GBM combines predictions from multiple weak learners, typically decision trees, to create a robust predictive model. GBM constructs predictive models by employing back-fitting and non-parametric regressions, continuously fitting new models through the minimization of a loss function, to generate the most accurate model possible. The ultimate goal of GBM is to find a functionF(x) [34], that minimizes its loss function L(y,F(x)) through iterative back-fitting:

$$F^* = \underset{F}{\operatorname{argmin}} E_{y,x} L(y, F(x)) \quad (5)$$

A boosted model is a weighted linear combination of base learners:

$$F(x; \{\beta_m, \theta_m\}_1^M) = \sum_{m=1}^{M} \beta_m h(x: \theta_m) \quad (6)$$

Here, $h(x: \theta)$ represents a base learner parameterized byθ.These base learners are considered weak learners because their individual predictions only marginally outperform random guessing. However, recursive learning with weak learners can achieve performance comparable to that of strong learning algorithms.

The weights β_m represent the importance or contribution of each base learner to the final model.

The goal of boosting methods, as discussed in [35], is to optimize the function space by iteratively constructing an estimated function F(x) in the additive form:

$$F(\mathrm{x}) = F^{I}(x) = \sum_{i=0}^{I} F_i(\mathrm{x}) \tag{7}$$

Here, I represents the number of iterations, F_0 is the initial guess, and $\{F_i\}_{i=1}^{I}$ denotes the function increments or boosts. Each F^i contributes to the ensemble prediction at iteration i. Typically, a base learner model $h(x,\theta)$ is selected to form these incremental functions, following a "greedy stage-wise" strategy.
Initially, the model initializes F_0 sets $h(x,\theta_0)$. Then, a new function $h(x,\theta_i)$ is chosen at each iteration to closely align with the negative gradient $g_i(x)$ along the observed data:

$$g_i(x) = E_y\left[\frac{\partial L(y,F(x))}{\partial F(x)}|x\right]_{F(x)=F^{i-1}(x)} \tag{8}$$

In the above equation, $L(.)$ represents the loss function. By assigning a step size ρ for each step, the new function increment is selected to be highly correlated with $g_i(x)$. Assuming we have N samples, and ρ_i is the line search on the direction of steepest-descent. the optimization problem can be solved using the classic least-squares minimization method:

$$(\rho_i,\theta_i) = \underset{\rho,\theta}{\operatorname{argmin}} \sum_{j=1}^{N}\left[-g_i(x_j) + \rho h(x_j,\theta)\right]^2 \tag{9}$$

The objective is to find the optimal values of ρ_i and θ_i that minimize the squared difference between the negative gradient and the base learner's prediction on the training data points x_j. This iterative process continues until the desired number of iterations I is reached.

3. METHODS

3.1 APPLYING THE SPIRIT OF GBM TO ENHANCE AE TRAINING

Many studies focus on improving the components of AEs to enhance imputation performance, often overlooking the significance of training data quality. However, when the training data is flawed, it inevitably constrains the efficacy of training. To attain a superior training dataset, leveraging insights from previous training efforts becomes imperative. A noteworthy approach for such reuse entails employing GBM.

In recent years, the concept of "boosting" has emerged as a promising direction to enhance model performance by leveraging ensemble techniques. It suggests that combining multiple models can yield better results than using a single model. Among various boosting methods, Gradient Boosting has demonstrated its effectiveness, especially in dealing with complex and high-dimensional datasets to achieve accurate predictions. The core idea behind Gradient Boosting is to sequentially train models, with each subsequent model focused on predicting the errors of the previous models. This iterative process is guided by a loss function that measures the quality of predictions and residuals. Commonly used loss functions in gradient boosting include Mean Squared Error (MSE) for regression problems and cross-entropy loss for classification problems.

While Gradient Boosting has been successfully applied in various domains, such as classification, decision forests, credit card fraud detection, and medical data analysis, [35-39], there have been limited works exploring its application in missing data imputation. Because GBM is conventionally employed in supervised learning scenarios where datasets are complete, its direct applicability to missing data is constrained by the inherent absence of certain information.

Nevertheless, drawing inspiration from the foundational principles of GBM, we can adapt and employ analogous strategies within our proposed methodology. Despite the inherent unsuitability of typical supervised learning approaches for datasets with missing values, we can extract insights from the essence of GBM to augment the efficacy of our proposed method. In the following sections, we will present the detailed methodology of our proposed approach and evaluate its performance through extensive experiments, comparing it with existing methods in the field.

3.2 GRADIENT BOOST MODULE IN AE FOR MISSING DATA IMPUTATION

Typically, GBM is not directly employed for training AEs, as these two fall under distinct machine learning paradigms. AEs are neural networks predominantly employed for unsupervised learning and dimensionality reduction. Nonetheless, the fundamental objective of GBM is to discover a function F(x) that minimizes its loss function L(y, F(x)) through iterative back-fitting. However, by drawing inspiration from the fundamental principles of GBM, we have the opportunity to adapt and implement similar strategies within the framework of our proposed methodology.

The fundamental concept of GBM implies that we can segment the entire AE training process into multiple stages. At each stage, training builds upon the achievements of the preceding stage, leading to an optimization process. In the domain of missing data imputation, where input data often contains gaps and irregularities, it becomes plausible to utilize data with enhanced reconstruction quality from one stage as the input for the subsequent stage. This essentially means that each stage of AE learning serves as a base learner, persistently learning and enhancing its performance iteratively. Consequently, this approach is expected to yield similar benefits in terms of data reconstruction and imputation.
Regarding the basic AE described in Section 3.1, it consists of an encoding function f(X) that transforms the input X into a code in the latent space. Z represented as Z = $W_e X + b_e$. The code Z is then transformed by a decoding function g(Z), which reconstructs the input as Y = $W_d Z + b_d$.

For a basic AE, the estimated function F(x) can be expressed as the composition of the encoding and decoding functions:

$$\mathrm{F}(x) = \mathrm{g}(\mathrm{f}(x)) \tag{10}$$

The parameters of the AE are denoted as $\theta = \{W_e, W_d, b_e, b_d\}$, and the loss function is the least-squares loss function, which can be expressed as $L(X,\mathrm{Y}) = \|X - Y\|^2$, where X represents the input data and Y is the reconstructed output.
In the boosting context, utilizing the least-squares loss, the optimization process can be expressed as:

$$(\rho_i, \theta_i) = \underset{\rho,\theta}{\operatorname{argmin}} \sum_{j=1}^{N} \left[y_j - (F_{j-1}(x_j)) + \rho h(x_j;\theta)\right]^2 \tag{11}$$

Here, y_j represents the target output, $F_{\mathrm{j}-1}(x_j)$ denotes the ensemble prediction from the previous iteration, which implies the previous iteration's reconstruction back-fitting to input X, $h(x_j;\theta)$ is the base learner's prediction, and ρ_i and θ_i are the optimal values of the step size and the base learner's parameters at the i-th iteration, To solve for F(x), a stage-wise model can be used:

$$F_i(x) = F_{i-1}(x) + \rho_i h(x;\theta_i) \tag{12}$$

Nonetheless,thepractical application of this concept necessitates empirical validation, appropriate dataset selection, parameter tuning, and thorough evaluation to ensure its effectiveness within specific domains and problem contexts. This staged AE training technique presents a promising approach for addressing challenges in missing data imputation while adhering to the iterative optimization philosophy of GBM. We propose two methods for implementing theGBM concept within AE for missing data imputation.

3.3. SHORT-TERM RECONSTRUCTION WITH ITERATIVE UPDATES (STR-IU) METHOD

The Short-Term Reconstruction with Iterative Updates (STR-IU) method entails the periodic direct replacement of training data X with the reconstructed values Y obtained from the AE. This approach assumes that the training of the prior base learner, which in AE corresponds to the previous training iteration, is effective and results in more informative reconstructions compared to the original input data. By substituting the missing input data with the improved reconstructions, the training process prioritizes the utilization of enhanced information from the preceding training iteration. To further refine the quality of the replacement data, a difference threshold parameter is employed.The parameters for applying theSTR-IU method in AE training experiments were determined as follows,

- Substitution Ratio (P_{SR})

P_{SR} was set to determine the proportion of the best-fit reconstructions that are replaced. P_{SR} governs the extent to which the training from the previous epoch contributes and is restricted within the range of $0 \leq P_{SR} \leq 1$. This constraint ensures a controlled updating process, progressively enhancing the overall ensemble prediction during the epochs of the reconstruction training interval. The choice of P_{SR} is crucial as it affects the balance between the original training data and the reconstructed data.

- Difference Threshold (P_{DT})

P_{DT} involves comparing the training results before and after optimization. This threshold ensures that replacement only occurs when the difference between the two exceeds a certain threshold value. Frequently replacing training data solely

to achieve lower RMSE values could lead to excessive and potentially unhelpful replacements, potentially resulting in an insufficient number of epochs for training. Setting a threshold value is essential to enhance the quality of replacement data and prevent overly frequent and dense replacements

- Number of Epochs in each replace iteration ($\boldsymbol{N_{period}}$).

This determines the duration of the training period, which is the number of epochs, allocated for the reconstruction process.
Please refer to Table 1 for a description of the STR-IU method algorithm.

Table2. The Algorithm of STR-IU Method

Algorithm 1: STR-IU method	
Require: Training dataset X, Difference Threshold $\boldsymbol{P_{DT}}$, Substitution Ratio $\boldsymbol{P_{SR}}$,Number of short training iteration $\boldsymbol{N_{period}}$,	
1	for Number of training iterations $\boldsymbol{N_{period}}$ do
2	Train AE with X to produce back fitting reconstruction y_{fit}
3	$when\ [err(X, y_{j+1}) - err(X, y_j)]$>$\boldsymbol{P_{DT}}$, y_{fit}=$(1 - \boldsymbol{P_{SR}})X + \boldsymbol{P_{SR}} y_{j+1}$
4	Until epoch=$\boldsymbol{N_{period}}$,take X as$F$(X), and byF_{j+1}(X) = $F_i(x) + \rho_i h(x; \theta_i)$
5	Substitutive X with P_{SR},$X_{iter+1} = (1 - P_{SR})X_{iter} + P_{SR} y_{fit\_iter}$
6	end for

The STR-IU approach entails replacing the missing values in the input data with the reconstruction values predicted by each of the previous training iterations, The fundamental concept of the STR-IU method is to continually update input predictions by integrating the reconstructions from each iteration, replacing missing values with imputed values, and minimizing the reconstruction error to enhance the quality of missing data imputation. Please refer to Figure 2 to visualize the AE training process of STR-IU.

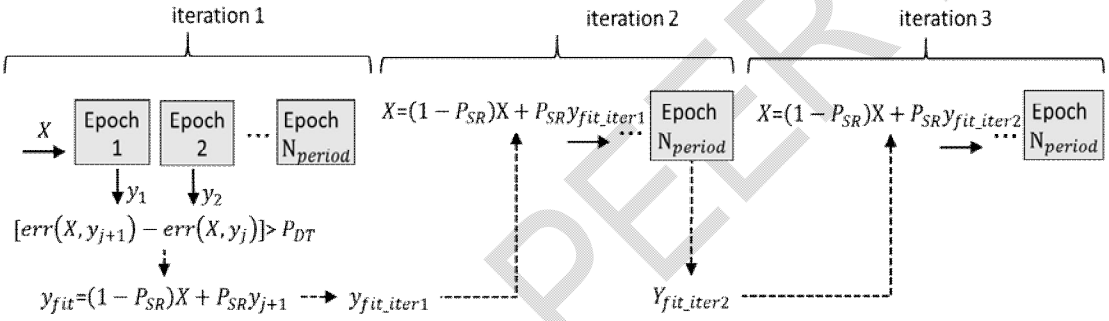


Figure2. STR-IU method in AE Training Process

3.4. LONG-TERM RECONSTRUCTION WITH A SINGLE UPDATE (LTR-SU) METHOD

The Long-Term Reconstruction with a Single Update (LTR-SU) method is focused on achieving the most accurate reconstruction of missing data within a reconstruction training interval to uncover underlying information. It also usesP_{SR} to determine the proportion of the best-fit reconstructions that are replaced. Subsequently, the most accurate reconstructionis merged with the original missing data using a specified recombination ratio P_{RR}. The recombination training data denoted as X_{remix}, is employed during the latter part of the training stage to create a consolidated model. The parameters for applying LTR-SU method in AE training experiments were determined as follows.

- Substitution Ratio (P_{SR})

Similar to theSTR-IU method.

- Recombination Ratio (P_{RR})

P_{RR} is to determine the proportion of the best-fit reconstructions that are recombined with the original training data once the best-fit reconstruction process is completed. P_{RR} influences how much emphasis is placed on the recombined data during training.

- Number of epochs for the best-fit training (N_{best_fit}).

N_{best_fit} is similar to N_{period}, It defines the duration of the training period. However, the reconstruction training period is longer, and it is trained only once.Please refer to Table 2 for a description of the LTR-SU method algorithm.

Table3. The Algorithm of LTR-SU Method

Algorithm 2: LTR-SU method
Require: Recombination Ratio $\boldsymbol{P_{RR}}$, Substitution Ratio $\boldsymbol{P_{SR}}$, Number of training epochs for fit reconstruction $\boldsymbol{N_{best\_fit}}$, Number of general training epochs $\boldsymbol{N}$,

Stage1:get fit reconstruction dataset y_{best}
for j in 1: N_{best_fit} do
Train AE with X to produce back itting reconstruction $y_{bestfit}$
when $\mathrm{err}(X, y_{j+1}) \leq \mathrm{err}(X, y_j), y_{bestfit} = (1 - P_{SR})y_{bestfit} + P_{SR}y_{j+1}$
end for
return fit reconstruction dataset $y_{bestfit}$
Stage 2: replace X to $X_{remix}$and start training
take X as$F(X)$,by $F_{j+1}(X) = F_{i-1}(x) + \rho_i h(x; \theta_i)$, and$X_{remix} = (1 - P_{RR})X + P_{RR}y_{bestfit}$
for j in N_{best_fit} + 1: N do
Train AE with X_{remix}
end for

The LTR-SU approach similarly aims to reconstruct training data from the original missing data. However, the core aspect of this method involves its execution over an extended period to achieve better reconstruction for training, followed by continuous training without altering the training data. Please refer to Figure 3 to visualize the AE learning process of theLTR-SU method

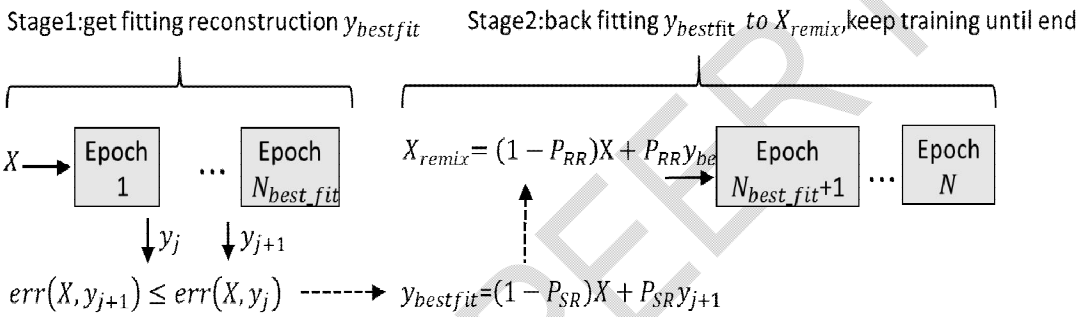


Figure 3. LTR-SU Method in AE Learning Process

4. EXPERIMENTS

4.1. DATASET DESCRIPTION

For the purpose of expedited and straightforward comparison between the experimental and control groups, we have opted for a dataset comprising exclusively numerical content. Simultaneously, to investigate whether the number of attributes and instances affects the training improvement outcomes, we aim to identify datasets with significant differences in these aspects. Finally, to confirm whether our proposed method is influenced by dataset characteristics and exhibits effectiveness solely for specific data types, we intend to select datasets from diverse application domains.

After conducting a thorough search and comparison, as shown in Table 1, our proposed approach is evaluated on three datasets sourced from the UCI Machine Learning Repository. The datasets used include the following:

- Cardiotocography Dataset

This dataset consists of measurements of fetal heart rate (FHR) and uterine contraction (UC) features on cardiotocograms, which were classified by expert obstetricians.

- Parkinson's Telemonitoring Dataset

This dataset includes a range of biomedical voice measurements from 42 individuals with early-stage Parkinson's disease who participated in a six-month trial of a telemonitoring device for remote symptom progression monitoring.

- Statlog (Landsat Satellite) Dataset

This dataset contains multi-spectral values of pixels in 3x3 neighborhoods in a satellite image. The attributes in this dataset are numerical, ranging from 0 to 255.
We have compiled certain data characteristics pertinent to the experiments in Table 3 below.

Table 4. Datasets

Dataset	Cardiotocography	Parkinsons Telemonitoring	Statlog(Landsat Satellite)
Instances	2126	5875	6435
Attributes	21	19	36
Attribute Type	Real	Integer, Real	Integer
Has Missing value	N	N	N
Offering Testing Dataset	N	N	Y

These datasets originate from diverse application domains, spanning from medical measurements to satellite imagery. As a result, each dataset possesses unique characteristics tailored to its specific domain of use. This diversity in data nature, ranging from healthcare-related measurements to satellite image features, provides a valuable set of challenges and opportunities for assessing the impact of dataset characteristics on the proposed methodology.

In addition, the datasets can be classified into two distinct groups based on the number of attributes: 23, 22, and 37. Similarly, the number of instances falls into two categories: 2126, 5875, and 6435. This classification serves as a structured basis for professional and academic comparison.

4.2. MODELS FOR COMPARISON

In our research, we primarily investigate the impact of updating input data during AE training, with a deliberate focus on avoiding extensive discussions related to optimization algorithms and loss functions. To assess the influence of modifying input data, we employ DAE models and examine how different optimization mechanisms affect the performance of our proposed methods. We choose DAE over VAE due to VAE's inclusion of a KL Divergence term, which remains unaffected by changes in input data.

In our iterative model enhancement processes$F_i(x) = F_{i-1}(x) + \rho_i h(x;\theta_i)$, the primary focus is on improving $F_i(x)$. Within the realm of AE, there exist numerous optimization algorithms well-suited to address the optimization of $\rho_i h(x;\theta_i)$. As part of our evaluation methodology, we have selected two classical optimization algorithms: Stochastic Gradient Descent (SGD) and Adam (Adaptive Moment Estimation) to serve as baseline models. These choices align with the rigorous standards of evaluation in machine learning and deep learning research. SGD and Adam are widely recognized and respected optimization techniques, and their use as benchmarks will provide a robust basis for assessing the performance of our proposed approach.

In order to streamline the subsequent figure presentations, abbreviated references for the relevant comparison methods have been established, as delineated in Table 4 below.

Table 5. Models for comparison

Model Description	**Abbreviation**
DAE with Original Stochastic Gradient Descent optimizer	SGD
Applying STR-IU method in DAE with SGD	STR-IU +SGD
Applying LTR-SU method in DAE with SGD	LTR-SU +SGD
DAE with Original Adaptive Moment Estimation optimizer	Adam
Applying STR-IU method in DAE with Adam	STR-IU + Adam
Applying LTR-SU method in DAE with Adam	LTR-SU + Adam

4.3. EXPERIMENTAL SETUP

After randomizing the original dataset, we partitioned it into training and testing datasets according to a specified ratio. Specifically, the first 70% of the dataset was allocated to the training data, while the remaining 30% constituted the testing data. Given that the original dataset is devoid of any missing or erroneous values, a subset of the original values within the training dataset is randomly substituted with -10 to signify missing values by a specified ratio. The choice of -10 is based on the values observed in three experimental datasets, thus resembling an outlier.

For the experiments, we used different corruption levels (5%, 15%, and 25%) on three datasets. The number of input dimensions was determined based on the number of attributes, and the encoding dimension was fixed at 12. We

initialized weights and biases with random values. Testing models were implemented using PyTorch, and data were scaled to a specific range between 0 and 1, based on the minimum and maximum values in the dataset.

The number of training epochs for the models ranged from 50 to 600, with intervals of 50. For network parameter tuning, we employed both the Adam optimizer and the SGD optimizer. In the case of SGD, we set a learning rate of 0.1, a momentum of 0.9, a batch size of 256, and used an MSE loss function. For Adam, we utilized a learning rate of 0.001, with other parameters set to their default values.

The RMSE metric is a commonly used measure for assessing model prediction accuracy, quantifying the square root of the mean of the squared differences between actual and predicted values. Lower RMSE values indicate better model performance. The calculation of RMSE is defined as follows: Given x_i as the actual value and y_i as the predicted value.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{N}(x_i - y_i)^2} \quad (13)$$

5. RESULTS AND DISCUSSION

5.1. APPLYING STR-IU METHOD IN DAE TRAINING

These parameter settings were chosen based on careful experimentation and observations, aiming to strike a balance between various factors influencing the STR-IU method AE learning process.

• Parameter Setting for Substitution Ratio (P_{SR})

The experimental results indicated that setting P_{SR} to either 1e-02 or 1e-04 had no significant effect. This lack of impact can be attributed to the fact that theSTR-IU method directly replaces a portion of the original data with the reconstruction data produced during training, aiming to enhance the quality of the training data for improved training performance. TheSTR-IU method relies on the quality of reconstruction data obtained at each iteration, which tends to be less stable. Therefore, a large-scale replacement may lead to interference and counterproductive results. As a result, we explored the range of P_{SR} values such as 1e-05, 1e-06, and 1e-07 to find suitable settings.

• Parameter Setting for Difference Threshold (P_{DT})

Determining an appropriate value for P_{DT} is a critical consideration. If the value is too small, it may not effectively achieve the threshold effect. Conversely, if the value is too large, it might result in a reduction in the number of replacements, which may also not be conducive to effective training. To address this, experiments were conducted on each of the three datasets using SGD, initially training for 200 epochs. The differences in RMSE between consecutive epochs were recorded and sorted in ascending order to observe the variations in numerical differences. The training was then repeated for 400 and 600 epochs, allowing for a comparison of how the numerical differences evolved with different epoch durations and datasets. A range of threshold values from 1e-04 to 3e-04 was adopted as they strike a balance between not being too large (resulting in too few replacements) and still exhibiting significant variability for observing the threshold effect. These values consistently ranked within the top 35-45% and 10-25% across different datasets and training durations.

• Parameter Setting for Number of Epochs in Iteration (N_{period})

In our experiments, we set values of 10 and 25 for these periods.

5.2. APPLYING LTR-SU METHOD IN DAE TRAINING

• Parameter Setting for Substitution Ratio (P_{SR})

Experiments were conducted on three different datasets, and effective P_{SR} values were found to be approximately in the range of 1e-02, 1e-04, and 1e-06.

• Parameter Setting for Recombination Ratio (P_{RR})

Similarly, after testing on three different datasets, P_{RR} values were found to be approximately in the range of 1e-04, and 1e-06.

• Parameter Setting for Number of Epochs for Best Fit Training (N_{best_fit})

To ensure the quality of reconstruction data, it is essential to set N_{best_fit} when the RMSE values has roughly converged. Referring to the RMSE convergence variations during the training of DAE for 200, 400, and 600 epochs, N_{best_fit} was set

to 30, 40, and 50. This choice of N_{best_fit} helps to achieve higher-quality reconstruction and contributes to the overall training process of the LTR-SU method.

The experimental parameters involve the application of the STR-IU and LTR-SU methods to three datasets, outlined in Table 5 and Table 6.

Table 6. Parameters for 3 Datasets in applying STR-IU method in DAE

	Cardiotoco		Parkinsons		Statlog	
	SGD	Adam	SGD	Adam	SGD	Adam
P_{SR}	1e-06	1e-06	1e-07	1e-05	1e-06	1e-06
P_{DT}	1e-04	3e-04	2e-04	3e-04	2e-04	1e-04
N_{period}	10	10	10	25	25	10

Table 7. Parameters for 3 Datasets in applying LTR-SU method in DAE

	Cardiotoco		Parkinsons		Statlog	
	SGD	Adam	SGD	Adam	SGD	Adam
P_{SR}	1e-02	1e-06	1e-02	1e-02	1e-06	1e-04
P_{RR}	1e-04	1e-04	1e-04	1e-06	1e-06	1e-06
N_{best_fit}	40	30	30	30	40	40

5.3 ISSUES ASSOCIATED WITH FINDING OPTIMAL PARAMETER COMBINATIONS.

In practical implementations, both SGD and Adam algorithms typically avoid fixing the random seed. While this practice contributes to improved learning performance in models, it introduces a challenge: even with identical initial values, the non-fixed nature of the random seed may result in slight numerical variations upon repeated executions. Although these differences may be negligible, they can become inconspicuous, especially during the process of searching for optimal parameters where the inherent variability of result values may not be apparent. In such circumstances, the fluctuation of random seeds can introduce subtle interference, potentially leading to misjudgments in the evaluation.

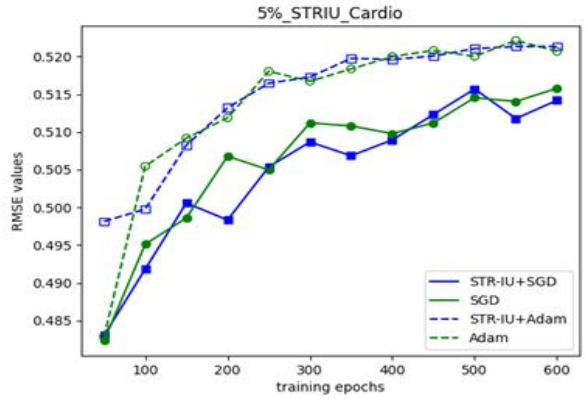
To mitigate this situation, we practice executing multiple consecutive runs using the same initial values and subsequently compute the average of these runs. This approach aims to reduce the impact of random seed variability on the results, ensuring greater stability and minimizing the potential for misinterpretation caused by stochastic influences.

Moreover, during the sequential search for optimal parameter combinations, there may be instances in which a specific parameter demonstrates favorable performance with two or three distinct values. While one approach is to consider all these values for comparison, doing so may significantly increase the time spent. Therefore, it is common to evaluate by observing the results of adjacent values. After all, the impact of parameters on the model is unlikely to be highly drastic. If a particular value performs well, the performance of adjacent values should also be relatively good. Thus, at times, comparing the performance of neighboring values is used to determine the optimal parameter value.

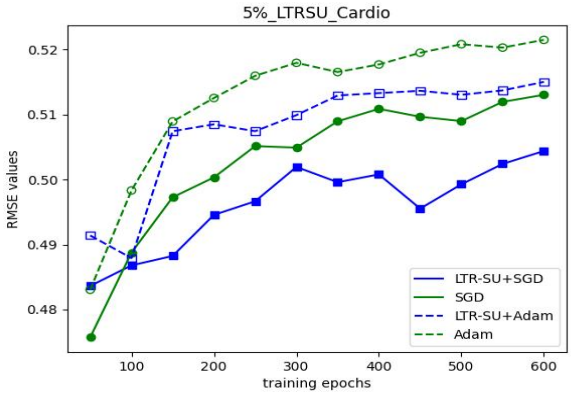
5.4 THE EXPERIMENT INVOLVES COMPARING 3 DATASETS USING THE RMSE METRIC

To compare the effectiveness of the STR-IU and LTR-SU methods, we divided the figures into two categories: (a) series depicting the differences between STR-IU and the SGD and Adam methods, while (b) series illustrating the disparities between LTR-SU and the SGD and Adam methods. We represented STR-IU+SGD and LTR-SU+SGD with solid squares connected by solid lines, SGD with solid circles connected by solid lines, STR-IU+Adam and LTR-SU+Adam with empty squares connected by dashed lines, and Adam with empty circles connected by dashed lines. Based on the experimental results of the three datasets at different missing data percentages, we generated a total of nine figures. First, the results for Cardio, Parkin, and Statio datasets with missing data at 5% are depicted in Figures 4, 5, and 6, respectively. Subsequently, the results for missing data at 15% are shown in Figures 7, 8, and 9, while the results for missing data at 25% are displayed in Figures 10, 11, and 12.

5.4.1 RMSE Value Variations for Each Dataset with 5% Missing Data

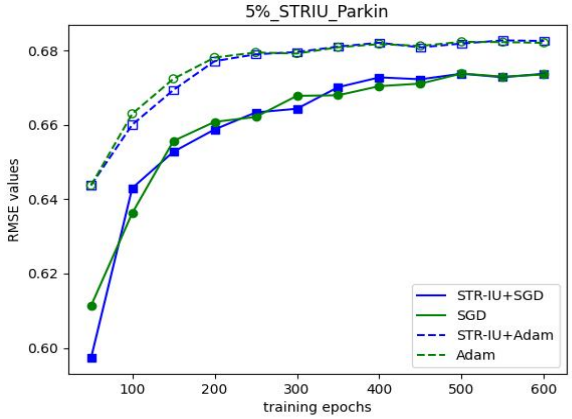


(a)

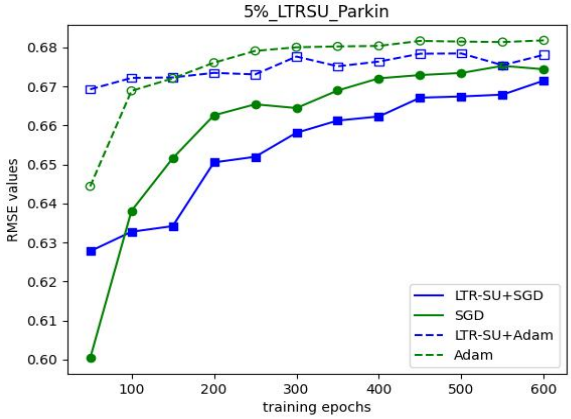


(b)

Figure 4. Comparison with SGD and Adam on Cardio Dataset with 5% Missing Data; (a) STR-IU Method; (b) LTR-SU Method

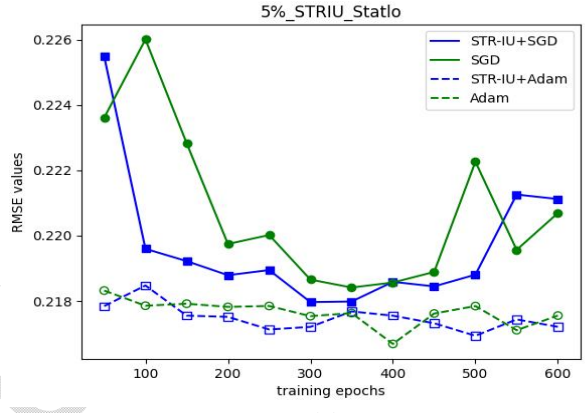


(a)

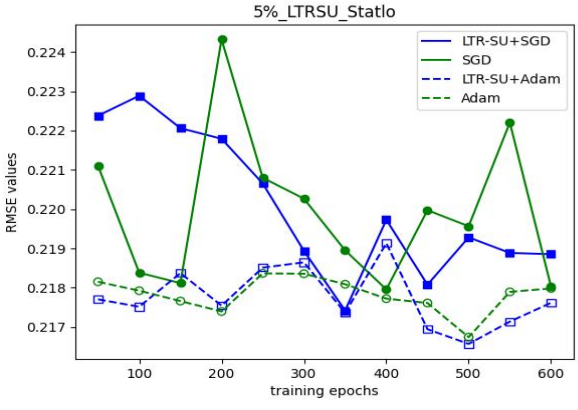


(b)

Figure 5. Comparison with SGD and Adam on Parkin Dataset with 5% Missing Data; (a) STR-IU Method; (b) LTR-SU Method



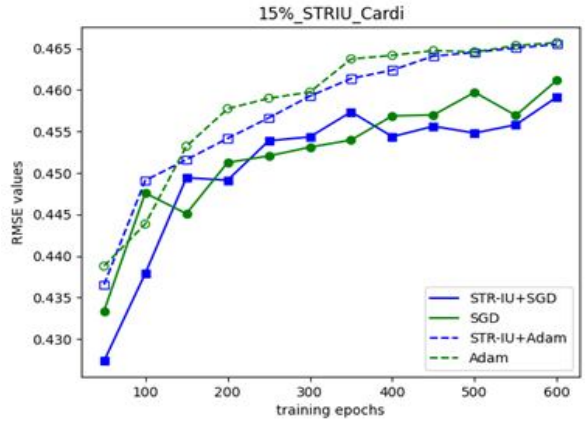
(a)



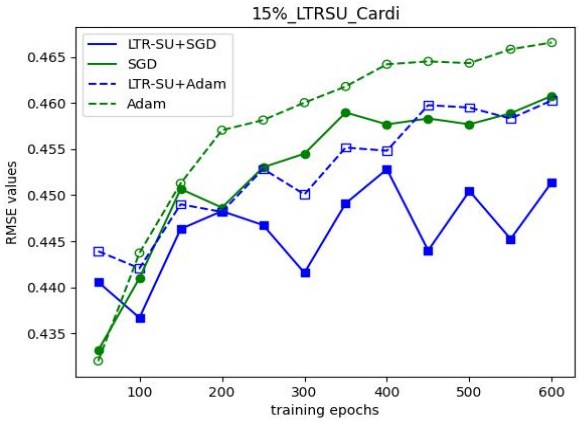
(b)

Figure 6. Comparison with SGD and Adam on Statio Dataset with 5% Missing Data; (a) STR-IU Method; (b) LTR-SU Method

5.4.2 RMSE Value Variations for Each Dataset with 15% Missing Data

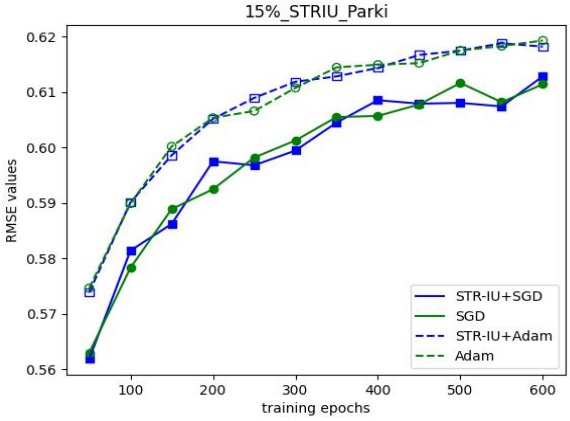


(a)

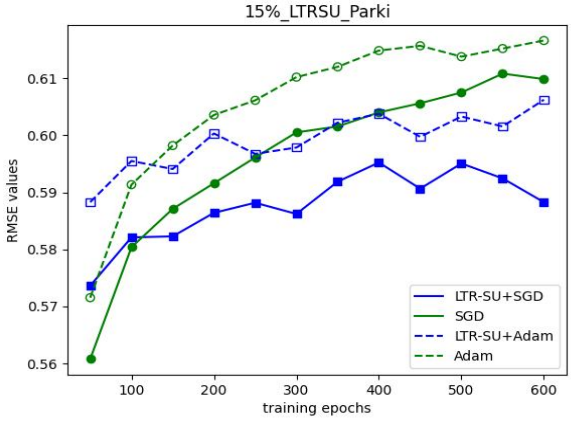


(b)

Figure 7. Comparison with SGD and Adam on Cardio Dataset with 25% Missing Data; (a) STR-IU Method; (b) LTR-SU Method

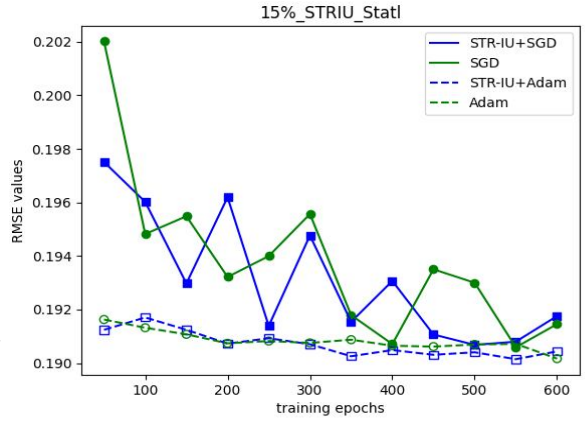


(a)

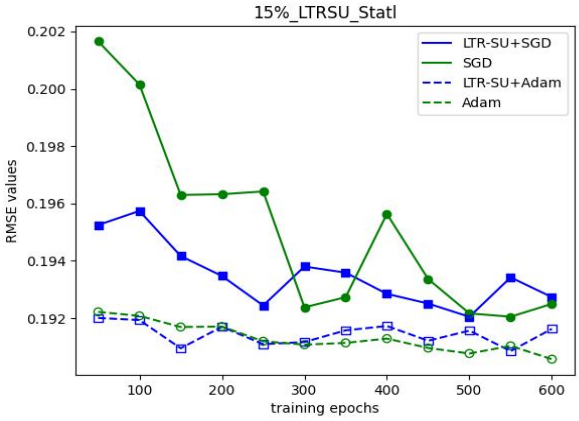


(b)

Figure 8. Comparison with SGD and Adam on Parkin Dataset with 25% Missing Data; (a) STR-IU Method; (b) LTR-SU Method



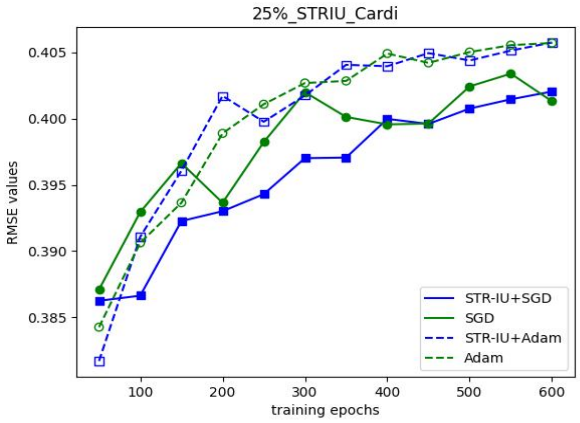
(a)



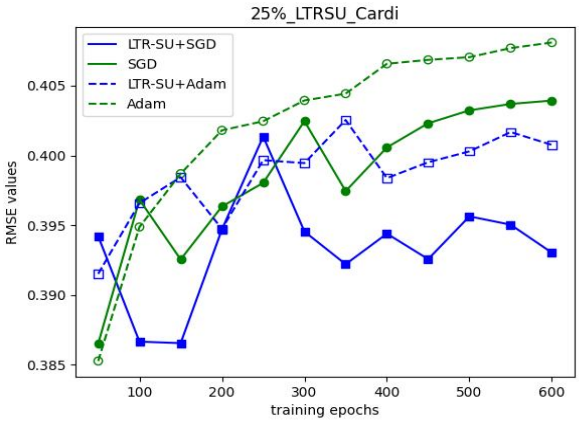
(b)

Figure 9. Comparison with SGD and Adam on Statio Dataset with 25% Missing Data; (a) STR-IU Method; (b) LTR-SU Method

5.4.3 RMSE Value Variations for Each Dataset with 25% Missing Data

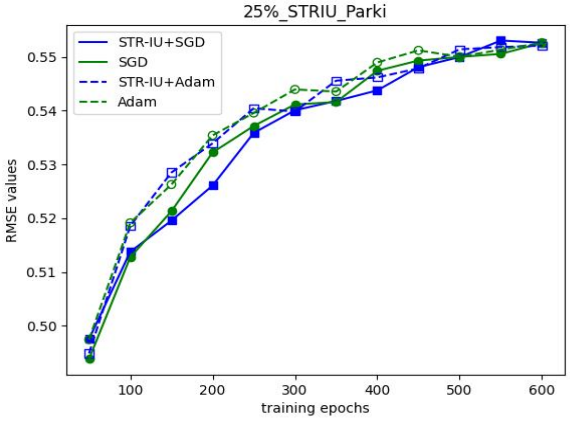


(a)

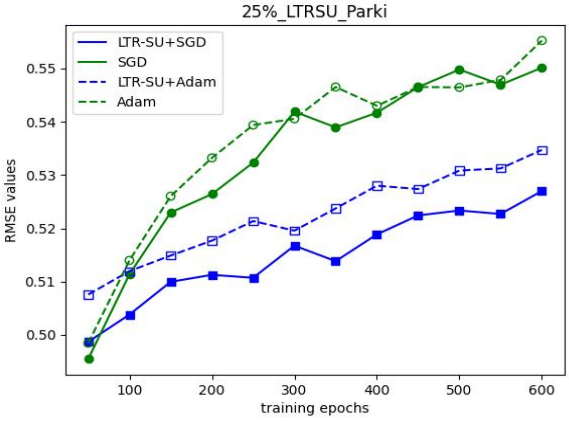


(b)

Figure 10. Comparison with SGD and Adam on Cardio Dataset with 25% Missing Data; (a) STR-IU Method; (b) LTR-SU Method



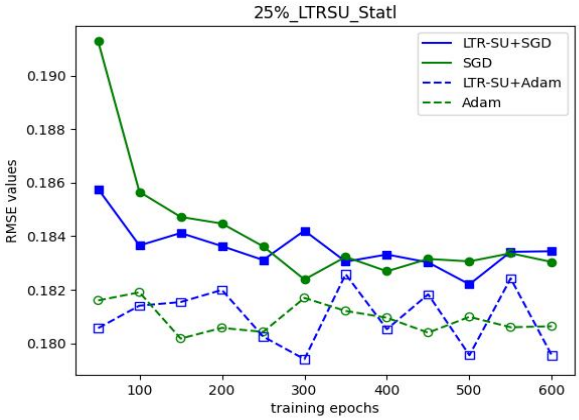
(a)



(b)

Figure 11. Comparison with SGD and Adam on Parkin Dataset with 25% Missing Data; (a) STR-IU Method; (b) LTR-SU Method

(a)



(b)

Figure 12. Comparison with SGD and Adam on Statio Dataset with 25% Missing Data; (a) STR-IU Method; (b) LTR-SU Method

In this experiment, LTR-SU consistently outperforms STR-IU, regardless of whether SGD or Adam is employed as the optimizer. Additionally, when comparing the use of SGD and Adam optimizers, it becomes evident that the improvements achieved with Adam are less pronounced compared to the enhancements observed with SGD.

In summary, LTR-SU may outperform STR-IU, especially when SGD is employed as the optimizer. This outcome can be rationalized by the fact that the reconstruction data refined by LTR-SU undergoes continuous optimization over an

extended period, leading to enhanced stability and accuracy when compared to the reconstruction data generated during the STR-IU iteration.

STR-IU's frequent and rapid data replacements can potentially compromise its stability. In the STR-IU approach, replacements occur whenever there is a reduction in the error of the reconstruction data, without taking into account its overall stability. Conversely, LTR-SU executes a single replacement with thoroughly optimized reconstruction data and allocates half of the epoch time for further training, which contributes to its improved performance.
When comparing the use of SGD and Adam optimizers, it is evident that, in comparison to the improvements seen with SGD, the enhancements achieved with Adam are less pronounced. This observation can be attributed to Adam's rapid convergence facilitated by its adaptive learning rates, which dynamically adjust based on parameter updates. Additionally, Adam's momentum term aids in accelerating the convergence process. While Adam generally performs well across various datasets, the advantages of training data optimization are not as prominent in this context.

Additionally, we observed that, in contrast to the Statlog dataset, higher numbers of epochs correspond to lower RMSE values, indicating better training effectiveness. However, for the Cardiotocography and Parkinson's datasets, higher numbers of epochs are associated with higher RMSE values. This discrepancy may stem from incomplete attribute distribution analysis during the sampling and creation of training and testing datasets for these two datasets. It is possible that the distribution characteristics of the training and testing datasets do not align perfectly, leading to the observed trend. Despite this potential limitation, our research method still demonstrates superior performance within the same dataset.

5.5 DISCUSSION

5.5.1 Effectiveness of LTR-SU and STR-IU Methods

The results show that the LTR-SU method generally outperforms the STR-IU method, particularly when SGD is used as the optimizer. This aligns with our hypothesis that leveraging iterative reconstruction updates (in LTR-SU) would lead to more stable and accurate imputation compared to frequent data replacements (in STR-IU). The findings support our research questions about the effectiveness of our proposed methods for enhancing AE training, indicating that LTR-SU provides better performance stability and accuracy in imputation tasks.

5.5.2 Comparison with Existing Optimization Techniques

While our research primarily focuses on improving AE training for missing data imputation through the integration of GBM principles, it aligns with previous studies that highlight the benefits of iterative optimization techniques in machine learning. The use of Adam and SGD optimizers as benchmarks is consistent with the standard practices in the field, ensuring that our findings are comparable with existing literature on optimization algorithms for neural networks.

5.5.3 Limitations and Future Directions

Our study acknowledges several limitations. First, the random seed variability in SGD and Adam can introduce subtle numerical variations, affecting the consistency of parameter search results. Second, the potential mismatch in attribute distribution between training and testing datasets, particularly for the Cardiotocography and Parkinson's datasets, might have influenced the RMSE values across different epochs. Lastly, the fixed selection of parameter values for P_{SR},P_{DT},P_{RR},N_{period} and N_{best_fit} may not capture the full spectrum of potential settings that could further enhance model performance.

Additionally, we only used the RMSE metric for evaluating model performance and did not consider the Mean Squared Error (MSE) metric. While RMSE is useful for penalizing larger errors more severely due to the square root operation, MSE provides a more straightforward measure of average error magnitude. The choice of metric can influence the interpretation of results, as RMSE tends to be more sensitive to outliers compared to MSE. Future studies should include both RMSE and MSE to provide a more comprehensive evaluation of imputation performance, considering their different applications and the insights they provide regarding model accuracy and error distribution.

5.5.4 Validation of Scheme Reliability

The reliability of our proposed methods was validated through extensive experimentation on three diverse datasets with varying levels of missing data. By analyzing the consistency of results across different runs and comparing the performance using RMSE, we ensured the robustness and stability of our methods. Furthermore, we conducted a

sensitivity analysis to understand the impact of different parameter settings on the model performance. The results show that our methods are reliable and can consistently improve imputation quality across various scenarios.

Future work should explore a broader range of parameter values and incorporate techniques to mitigate random seed variability. Additionally, incorporating multiple evaluation metrics can offer a more nuanced understanding of model performance and robustness in different scenarios

6. CONCLUSION

DAEs and VAEs represent distinct types of AEs, each with unique training objectives and architectural characteristics. DAEs are primarily designed for data denoising and feature learning tasks, whereas VAEs focus on probabilistic modeling and generative capabilities.

In our research, we primarily explored the implications of updating input data during AE training while avoiding extensive discussions on optimization algorithms and loss functions. To assess the impact of altering input data, we utilized DAE models and examined how various optimization mechanisms affected our proposed methods. We conducted comprehensive comparisons between SGD and the Adam optimization algorithm. We opted for DAE over VAE due to VAE's incorporation of a KL Divergence term, which remains unaffected by changes in input data.

Therefore, for our experiments, we selected the DAE architecture and considered two commonly used optimizers, namely SGD and Adam.Through our experiments, we demonstrated that both LTR-SU and STR-IU approaches can significantly enhance the training quality of the base model. LTR-SU involves accumulating training results over stages and then performing a single replacement, while STR-IU replaces original training data directly based on each training iteration's results. Both approaches have proven to be effective.

In our experiments, we transformed three distinct datasets into synthetic datasets with varying levels of missing data (5%, 15%, 25%). The summarized results indicate that, while performance may not consistently excel across all training epoch settings, there is a noticeable overall improvement when updating input data, whether using SGD or Adam. Additionally, LTR-SU outperforms STR-IU, and models with DAE using SGD exhibit greater optimization compared to DAE using Adam.

While it is acknowledged that parameter selection often necessitates experimentation and fine-tuning, empirical evidence from our research indicates that parameter ranges suitable for addressing missing data do not exhibit substantial variations across distinct datasets and optimization techniques. This observation underscores the robustness and transferability of the proposed method. Consequently, identifying appropriate parameter configurations does not pose a formidable challenge.

The significance of this finding lies in its potential to streamline the process of handling missing data effectively. It offers a standardized approach that can be applied with confidence across diverse datasets and analytical contexts. By simplifying parameter selection, this research contribution enhances the efficiency and reliability of missing data treatment, benefiting a wide range of data analysis and machine learning applications.

In our future research, we aim to explore other potentially valuable parameters. For instance, we plan to investigate whether excluding training data during the early stages of AE training, when convergence is still substantial, can improve subsequent replacement outcomes. Additionally, even within the same dataset, the training epoch at which the best performance is achieved may vary. Thus, identifying the optimal training epoch remains a subject of interest for future study

Disclaimer (Artificial intelligence)

Option 1:

Author(s) hereby declare that NO generative AI technologies such as Large Language Models (ChatGPT, COPILOT, etc) and text-to-image generators have been used during writing or editing of manuscripts.

Option 2:

Author(s) hereby declare that generative AI technologies such as Large Language Models, etc have been used during writing or editing of manuscripts. This explanation will include the name, version, model, and source of the generative AI technology and as well as all input prompts provided to the generative AI technology

Details of the AI usage are given below:

1.

2.

3.

REFERENCES

1. BROWN, Marvin L.; KROS, John F. Data mining and the impact of missing data. Industrial management & data systems, 2003, 103.8: 611-621. Author 1 AB, Author 2 CD. Title of the chapter. In: Title of the Book, 2nd ed. Publisher name; 2023. pp. 102–144.

2. FAN, Jianqing; HAN, Fang; LIU, Han. Challenges of big data analysis. National science review, 2014, 1.2: 293-314.

3. GRAHAM, John W.; CUMSILLE, Patricio E.; ELEK-FISK, Elvira. Methods for handling missing data. Handbook of psychology, 2003, 87-114.

4. ZHANG, Zhongheng. Missing values in big data research: some basic skills. Annals of translational medicine, 2015, 3.21.

5. KWAK, Sang Kyu; KIM, Jong Hae. Statistical data preparation: management of missing values and outliers. Korean journal of anesthesiology, 2017, 70.4: 407.

6. BARALDI, Amanda N.; ENDERS, Craig K. An introduction to modern missing data analyses. Journal of school psychology, 2010, 48.1: 5-37.

7. RUBIN, Donald B. Inference and missing data. Biometrika, 1976, 63.3: 581-592.

8. LITTLE, Roderick JA; RUBIN, Donald B. Statistical analysis with missing data. John Wiley & Sons, 2019.

9. OSMAN, Muhammad S.; ABU-MAHFOUZ, Adnan M.; PAGE, Philip R. A survey on data imputation techniques: Water distribution system as a use case. IEEE Access, 2018, 6: 63279-63291.

10. MADLEY-DOWD, Paul, et al. The proportion of missing data should not be used to guide decisions on multiple imputation. Journal of clinical epidemiology, 2019, 110: 63-73.

11. SCHAFER, Joseph L. Multiple imputation: a primer. Statistical methods in medical research, 1999, 8.1: 3-15.

12. ALICE, M. Imputing missing data with R; MICE package 2015. 2018.

13. CHEEMA, Jehanzeb R. Some general guidelines for choosing missing data handling methods in educational research. Journal of Modern Applied Statistical Methods, 2014, 13.2: 3.

14. PEREIRA, Ricardo Cardoso, et al. Reviewing autoencoders for missing data imputation: Technical trends, applications and outcomes. Journal of Artificial Intelligence Research, 2020, 69: 1255-1285.

15. VINCENT, Pascal, et al. Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning. 2008. p. 1096-1103.

16. RYU, Seunghyoung; KIM, Minsoo; KIM, Hongseok. Denoising autoencoder-based missing value imputation for smart meters. IEEE Access, 2020, 8: 40656-40666.

17. LU, Haw-minn; PERRONE, Giancarlo; UNPINGCO, José. Multiple imputation with denoising autoencoder using metamorphic truth and imputation feedback. arXiv preprint arXiv:2002.08338, 2020.

18. ZHANG, Jianye; YIN, Peng. Multivariate time series missing data imputation using recurrent denoising autoencoder. In: 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2019. p. 760-764.

19. KINGMA, Diederik P.; WELLING, Max. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

20. REZENDE, Danilo Jimenez; MOHAMED, Shakir; WIERSTRA, Daan. Stochastic backpropagation and approximate inference in deep generative models. In: International conference on machine learning. PMLR, 2014. p. 1278-1286.

21. ROSKAMS-HIETER, Breeshey; WELLS, Jude; WADE, Sara. Leveraging variational autoencoders for multiple data imputation. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Cham: Springer Nature Switzerland, 2023. p. 491-506.

22. XIE, Ruimin, et al. Supervised variational autoencoders for soft sensor modeling with missing data. IEEE Transactions on Industrial Informatics, 2019, 16.4: 2820-2828.

23. MAKHZANI, Alireza, et al. Adversarial autoencoders. arXiv preprint arXiv:1511.05644, 2015.

24. TOLSTIKHIN, Ilya, et al. Wasserstein auto-encoders. arXiv preprint arXiv:1711.01558, 2017.

25. VINCENT, Pascal, et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research, 2010, 11.12.

26. NG, Andrew, et al. Sparse autoencoder. CS294A Lecture notes, 2011, 72.2011: 1-19.

27. GAO, Fuchang. Ae^{2}I: A Double Autoencoder for Imputation of Missing Values. arXiv preprint arXiv:2301.06633, 2023.

28. LIU, Xin; ZHANG, Zijun. A two-stage deep autoencoder-based missing data imputation method for wind farm SCADA data. IEEE Sensors Journal, 2021, 21.9: 10933-10945.

29. LU, Xugang, et al. Ensemble modeling of denoising autoencoder for speech spectrum restoration. In: Interspeech. 2014. p. 885-889.

30. WANG, Yanxia, et al. Missing data imputation with OLS-based autoencoder for intelligent manufacturing. IEEE Transactions on Industry Applications, 2019, 55.6: 7219-7229.

31. MIOK, Kristian, et al. Multiple imputation for biomedical data using monte carlo dropout autoencoders. In: 2019 E-Health and Bioengineering Conference (EHB). IEEE, 2019. p. 1-4.

32. YE, Yongchao; ZHANG, Shuyu; JAMES, J. Q. Traffic data imputation with ensemble convolutional Autoencoder. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE, 2021. p. 1340-1345.

33. FRIEDMAN, Jerome H. Greedy function approximation: a gradient boosting machine. Annals of statistics, 2001, 1189-1232.

34. HE, Zhiyuan, et al. Gradient boosting machine: a survey. arXiv preprint arXiv:1908.06951, 2019.

35. DONG, Manqing, et al. GrCAN: gradient boost convolutional autoencoder with neural decision forest. arXiv preprint arXiv:1806.08079, 2018.

36. SEYFIOĞLU, Mehmet Saygın; ÖZBAYOĞLU, Ahmet Murat; GÜRBÜZ, Sevgi Zubeyde. Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities. IEEE Transactions on Aerospace and Electronic Systems, 2018, 54.4: 1709-1723.

37. RUSHIN, Gabriel, et al. Horse race analysis in credit card fraud—deep learning, logistic regression, and Gradient Boosted Tree. In: 2017 systems and information engineering design symposium (SIEDS). IEEE, 2017. p. 117-121.

38. VO, Minh Thanh; NGUYEN, Trang; LE, Tuong. Robust head pose estimation using extreme gradient boosting machine on stacked autoencoders neural network. IEEE Access, 2019, 8: 3687-3694.

39. JANG, Jong-Hwan, et al. Unsupervised feature learning for electrocardiogram data using the convolutional variational autoencoder. PLoS One, 2021, 16.12: e0260612.

40.Song j, yang z, li x. Missing data imputation model for dam health monitoring based on mode decomposition and deep learning. Journal of civil structural health monitoring. 2024 mar 5:1-4.

DEFINITIONS, ACRONYMS, ABBREVIATIONS

Here is the Definitions section. This is an optional section.

Term: Definition for the term