Master's Thesis Presentation for the 141th Dissertation Evaluation

# A Study of Improving Neural Missing Value Imputation by Chain Approach

## 체인 접근법을 통한 신경망 기반 결측치 보간법에 관한 연구

Hyejin Baek

bhj4208@gmail.com

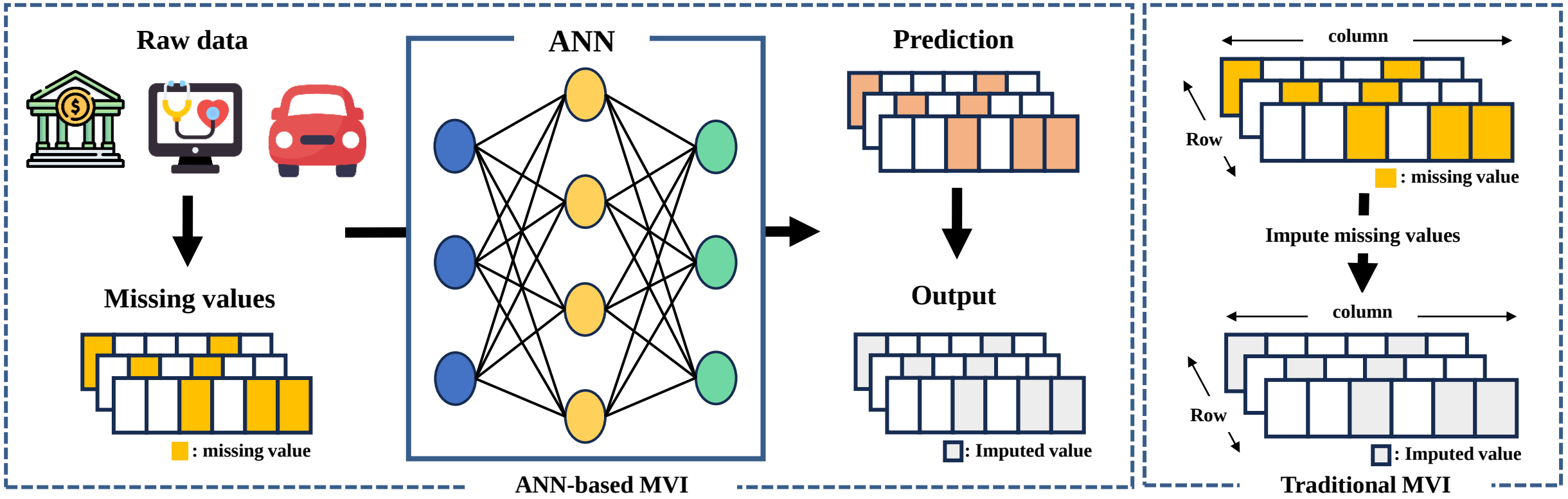
21th May, 2024

# Contents

# Introduction

## ANN-based Missing Value Imputation

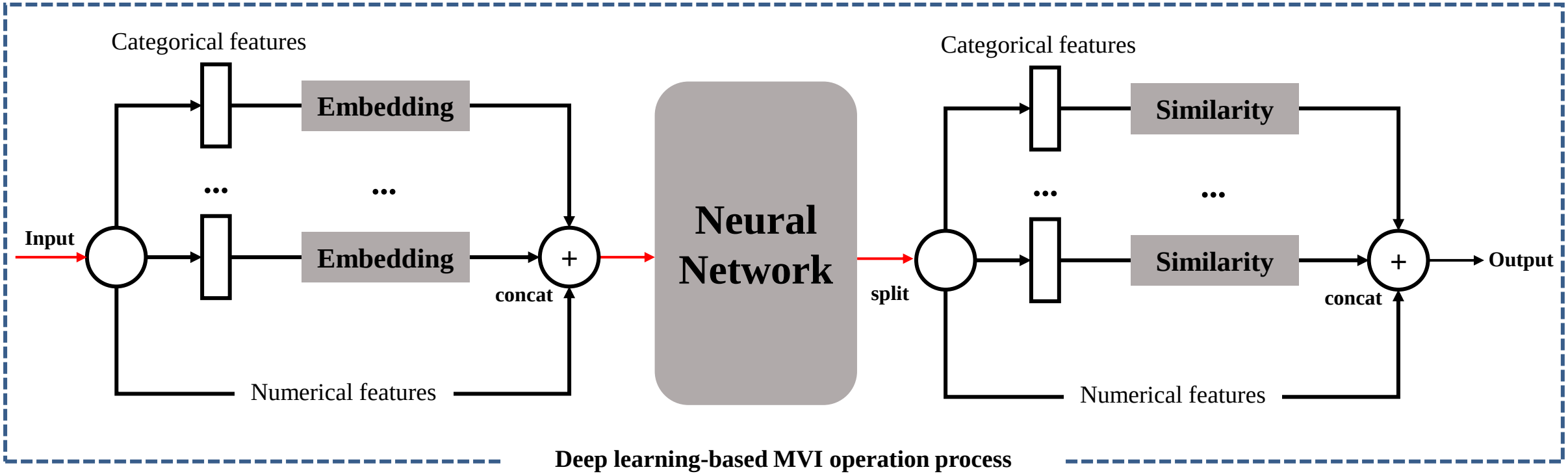
- Missing Value Imputation(MVI) is crucial in various fields, such as finance, medicine, etc, to enhance model accuracy by replacing missing values. It helps maintain data completeness and improves predictions in domains where missing values can bias results.
- In recent years, Artificial Neural Networks (ANN) have emerged as a powerful tool for MVI. Unlike traditional statistical methods like kNN, ANN can identify and train on more complex patterns and relationships in the data.



# Introduction

## Objective and Procedure

- MVI's research objective is to develop and validate methods for accurately replacing missing values(Not-a-Number, NaN), thereby enhancing the completeness and reliability of datasets used in data analysis and modeling.
- The conventional method uses separate models for each feature value, while the chain-based method sequentially trains models for each feature, stacking the imputed results to complete the dataset.

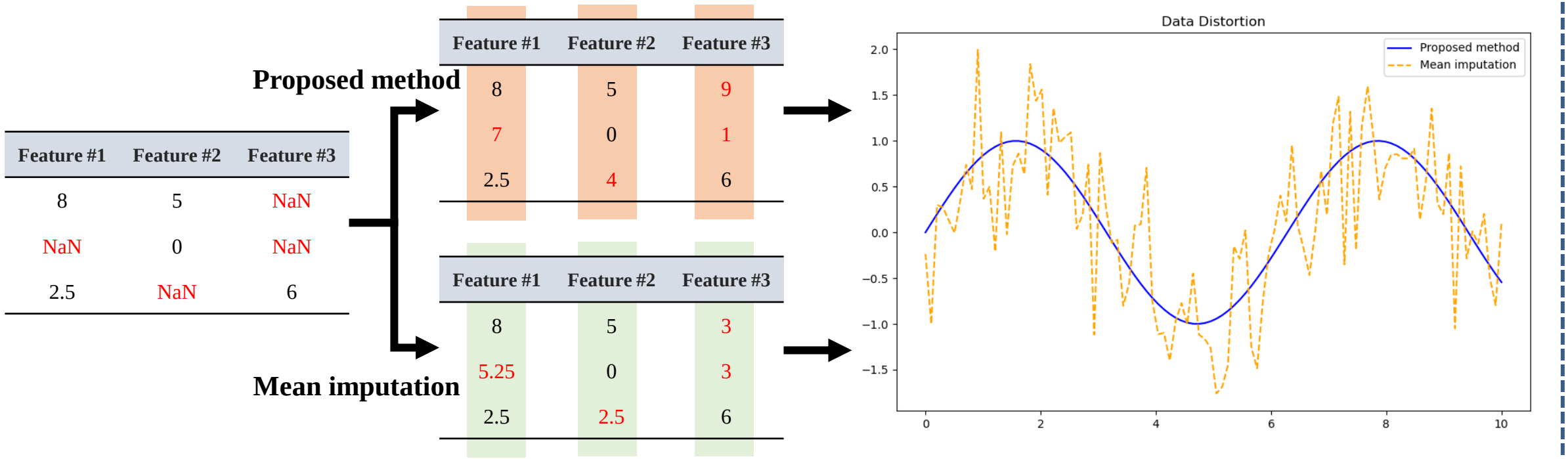



**Deep learning-based MVI operation process**

# Introduction

## Problem and Strategy

- The traditional MVI method typically employs simple calculations, such as replacing missing values with averages. ANN-based MVI utilizes the correlations between complex data structures and patterns to impute values similar to the actual data.
- The goal of this study is to prevent data distortion and impute missing values with reliable data by generating models for each feature and conducting ANN-based imputation through a chaining strategy.

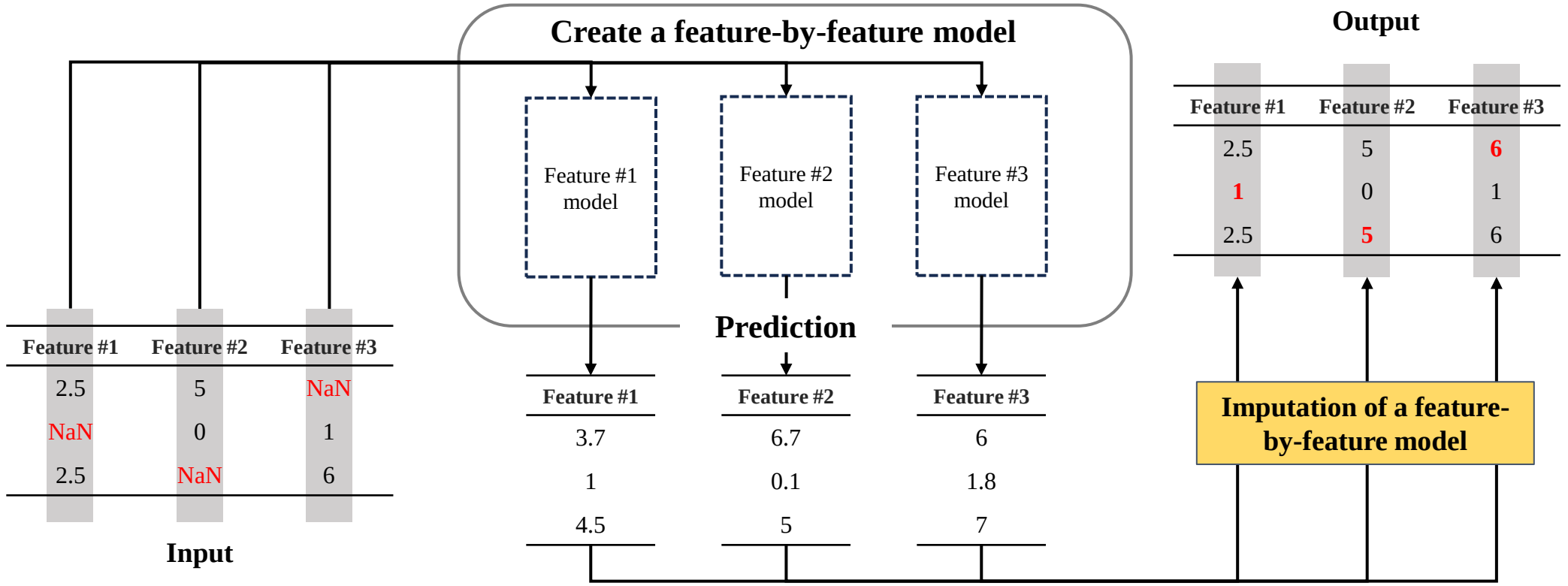



**Comparison of data distortions according to the imputation method**

# Related Work

## DataWig(DW) [1]

- DataWig(DW) creates a model for each feature to perform imputation. It aims to estimate the probability of all potential features for a feature given an imputation model and information from other features, especially those containing NaN values.
- Accurately capturing the full range of data characteristics can be challenging in datasets with many NaN values. As a result, the model may need help fully understanding and managing missing features.

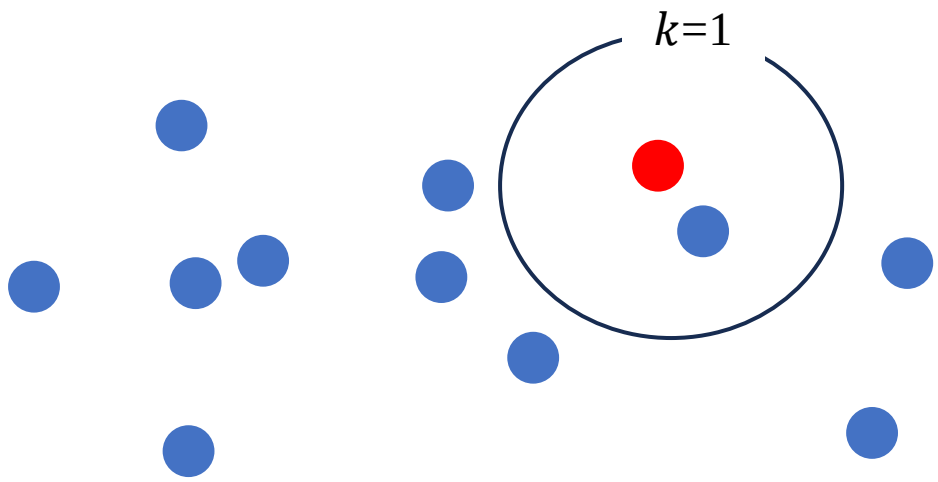



[1] Biessmann, Felix, et al. "DataWig: Missing value imputation for tables." Journal of Machine Learning Research 20.175 (2019): 1-6.

# Related Work

## $k$-Nearest Neighbor($k$NN) imputation [2]

- The $k$-Nearest Neighbor($k$NN) imputation replaces missing values by utilizing the nearest neighbor observation value $k$ for data points with missing values. This enhances the completeness of the dataset and provides valuable information for analysis.
- When missing features are replaced using the nearest neighbor $k$, inaccuracies may occur if the missing features are complex or unique. Additionally, locating neighbors for imputation can be challenging if the data distribution is highly imbalanced.




1. Determine the parameter $k$ = 1
2. Calculate the Euclidian distance between missing data instances and complete data

$$d_{(x,y)} = \sqrt{\sum_{j=1}^{s} (x_j - y_j)^2}$$

3. The value imputation is calculated using Weight Mean Estimation in equation

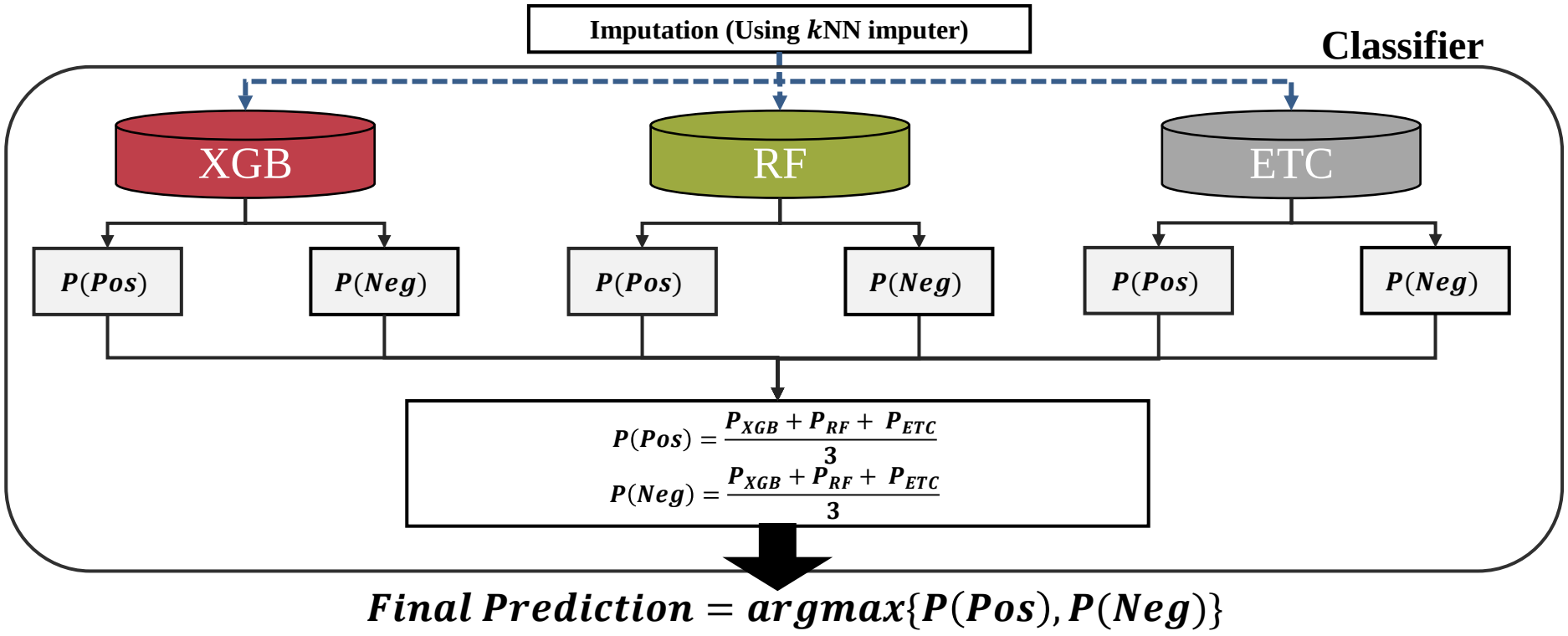
$$W_J = \frac{\sum_{k=1}^{K} W_k v_k}{\sum_{k=1}^{K} W_k}$$

[2] D. M. P. Murti, U. Pujianto, A. P. Wibawa and M. I. Akbar, "K-Nearest Neighbor (K-NN) based Missing Data Imputation," 2019 5th International Conference on Science in Information Technology (ICSITech), Yogyakarta, Indonesia, 2019, pp. 83-88

# Related Work

## Stacked Ensemble(SE) [3]

- 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.
- Combining multiple different models to construct a stacking ensemble can increase the complexity of the model. As the model becomes more complex, it may overfit the training data, decreasing its ability to generalize to new data.




[3] Turki Aljrees, Improving prediction of cervical cancer using KNN imputer and multi-model ensemble learning. PLOS ONE 19(1) 2024: e0295632.

# Related Work

## Common Drawback of Conventional Methods

- The table below summarizes the authors, years, references, and brief descriptions of the respective methods of the conventional method, along with descriptions of their drawbacks.
- Conventional methods impute missing values separately for each feature, which increases the likelihood of data distortion, resulting in a performance decline.

| Method | Name / Year | Reference | Summary | Drawback |
|---|---|---|---|---|
| DW | Biessmann / 2019 | Journal of Machine Learning Research | • Creates a model for each feature. | • Fewer training features due to NaN values |
| *k*NN | D. M. P. Murti / 2019 | ICSITech | • Replaces missing values with values from the most similar neighboring data points | • Inaccuracies may occur when features of missing values are complex or unique. |
| SE | Aljrees / 2024 | PLOS ONE | • It predicts a stacked ensemble classifier with Extreme Gradient Boosting(XGB), Random Forest(RF), and Extra Tree Classifier(ETC) models | • If each model is overfitting, it can degrade the performance of the final prediction. |

# Proposed Method

## Notation and Definition

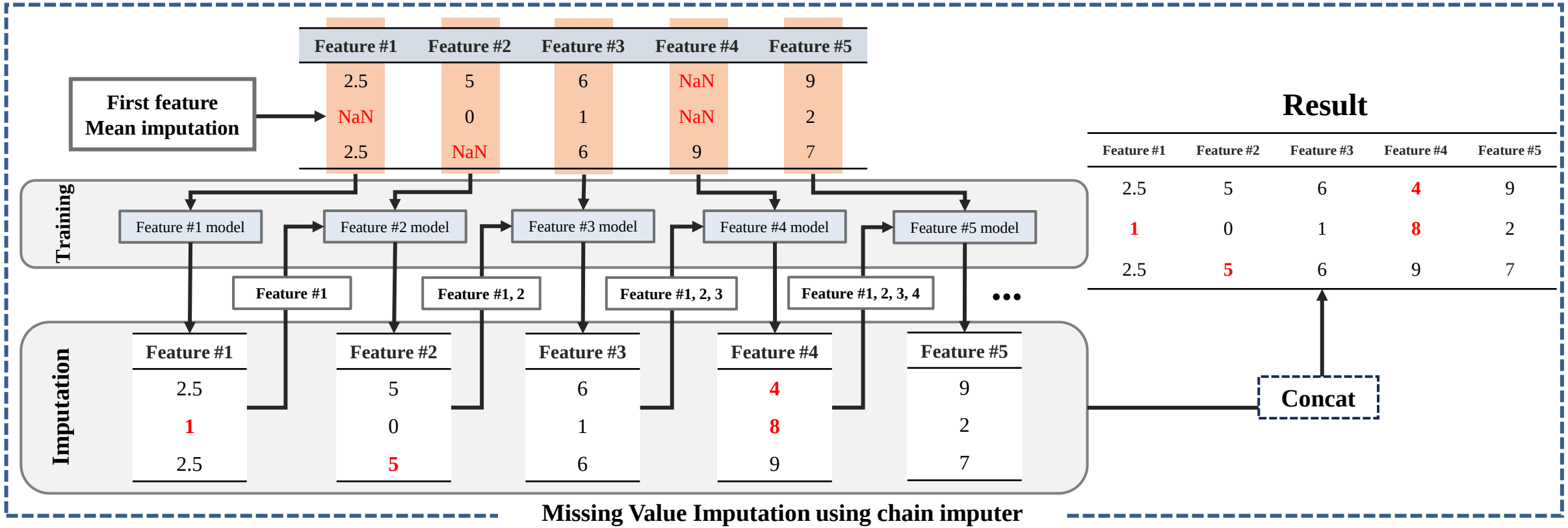
- The $\mathcal{D}$ is a set of datasets, and each dataset consists of an input vector $x_i$ and a label $y_i$ for the corresponding input. That is, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{f}$. The $f$ is feature representing one individual observation in the dataset. That is, it has several features.
- MVI is filling in missing values in the dataset($\mathcal{D}_{missing}$). It selects input features based on the given list of features(*train_col*) and trains an imputation model, named $M$ to predict missing values using the selected input features.

| Symbol | Name | Description |
|---|---|---|
| $\mathcal{D}$ | Dataset | Entire dataset |
| $x$ | Input | Input features or independent variables |
| $y$ | Output | Output or dependent variable |
| $f$ | Number of features | Total number of features |
| $\mathcal{D}_{missing}$ | Dataset with missing values | Dataset containing missing values |
| $\mathcal{D}_{imputed}$ | Dataset by filling missing values | Datasets without missing values |
| $M$ | Imputation model | Model used for imputation |

# Proposed Method

## Procedural Overview

- The model is constructed using the data from each feature individually. When moving to the next feature, it is trained along with the imputed results from the previous feature.
- This process is repeated sequentially for each feature, creating independent and sophisticated models for each feature of the data. This method allows for effective imputation of missing values and enhances the overall quality of the models.

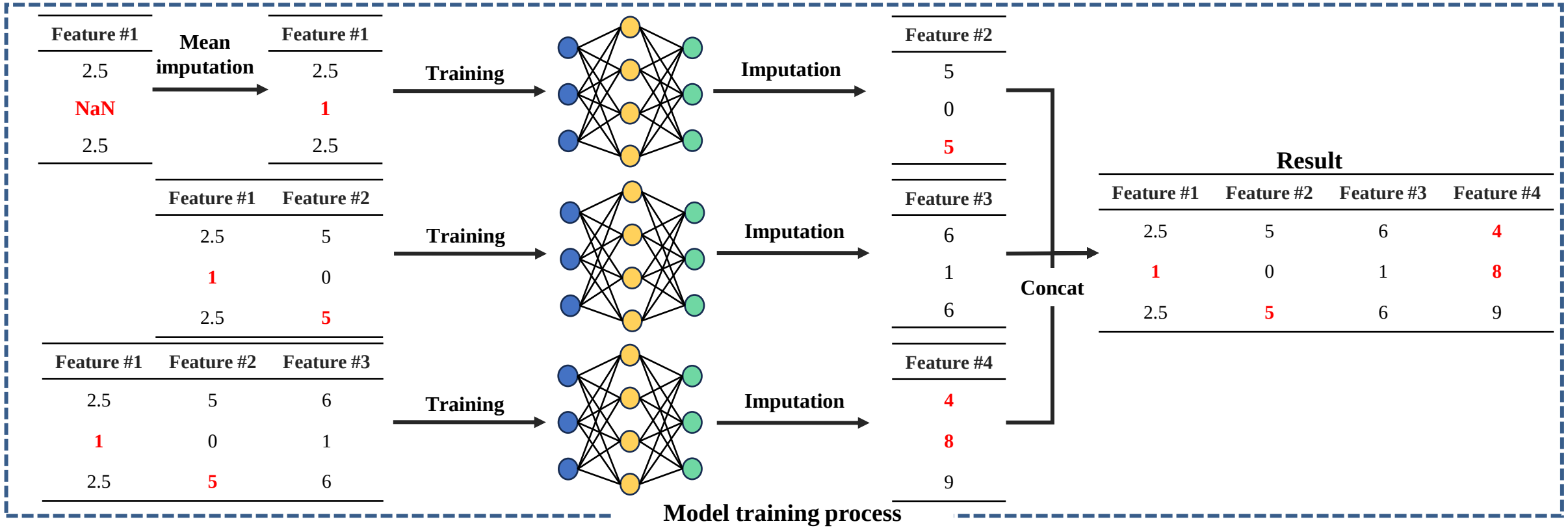



Missing Value Imputation using chain imputer

# Proposed Method

## Model training by feature

- The proposed method iteratively trains models to handle missing values, utilizing previously filled missing values to train each new model and replacing all missing values in the dataset.
- A critical aspect of the proposed approach is establishing a chain for training feature-wise models. This method trains a model using the current input features and leverages it to replace missing values, ensuring a robust and reliable data analysis process.



# Proposed Method

## Pseudo code

- The following pseudocode presents a method for imputing missing values in a dataset. It starts by utilizing features without missing values and then imputes missing values for required features.
- Input and output : The input is the entire dataset $\mathcal{D}$ with missing values, and the output is the imputed dataset $D_{imputed}$. The method aims to fill in the missing values in $\mathcal{D}$ to generate $D_{imputed}$, ensuring a complete dataset.

**Algorithm 1** Procedures of proposed imputation method

**Input** : Entire dataset $D$ (with missing values)

**Output:** Imputed dataset $D_{\text{imputed}}$

$S \leftarrow \emptyset$

**for** *each feature* $f$ *in* $D$ **do**

  **if** $f$ *has no missing values* **then**

    $S \leftarrow S\{f\}$

**if** $S = \emptyset$ **then**

  Choose the first feature $f \in F$

  $f' \leftarrow meanImputer(f)$

  $F = F \setminus \{f\}$

  $S \leftarrow S \cup \{f\}$

**for** *each feature* $f$ *in* $F$ **do**

  $f' \leftarrow M(S, f)$

  $F = F \setminus \{f\}$

  $S \leftarrow S \cup \{f\}$

# Proposed Method

## Pseudo code

- Initialize an empty set S : The method starts by initializing an empty set S to contain features without missing values. This step ensures that S initially holds features that do not require imputation.
- For each feature $f$ in $\mathcal{D}$, if $f$ has no missing values, it is added to S. This ensures that S initially comprises features that do not require imputation.

**Algorithm 1** Procedures of proposed imputation method

**Input** : Entire dataset $D$ (with missing values)

**Output:** Imputed dataset $D_{\text{imputed}}$

$S \leftarrow \emptyset$
**for** *each feature* $f$ *in* $D$ **do**
  **if** $f$ *has no missing values* **then**
    $S \leftarrow S\{f\}$
**if** $S = \emptyset$ **then**
  Choose the first feature $f \in F$
  $f' \leftarrow meanImputer(f)$
  $F = F \setminus \{f\}$
  $S \leftarrow S \cup \{f\}$
**for** *each feature* $f$ *in* $F$ **do**
  $f' \leftarrow M(S, f)$
  $F = F \setminus \{f\}$
  $S \leftarrow S \cup \{f\}$

# Proposed Method

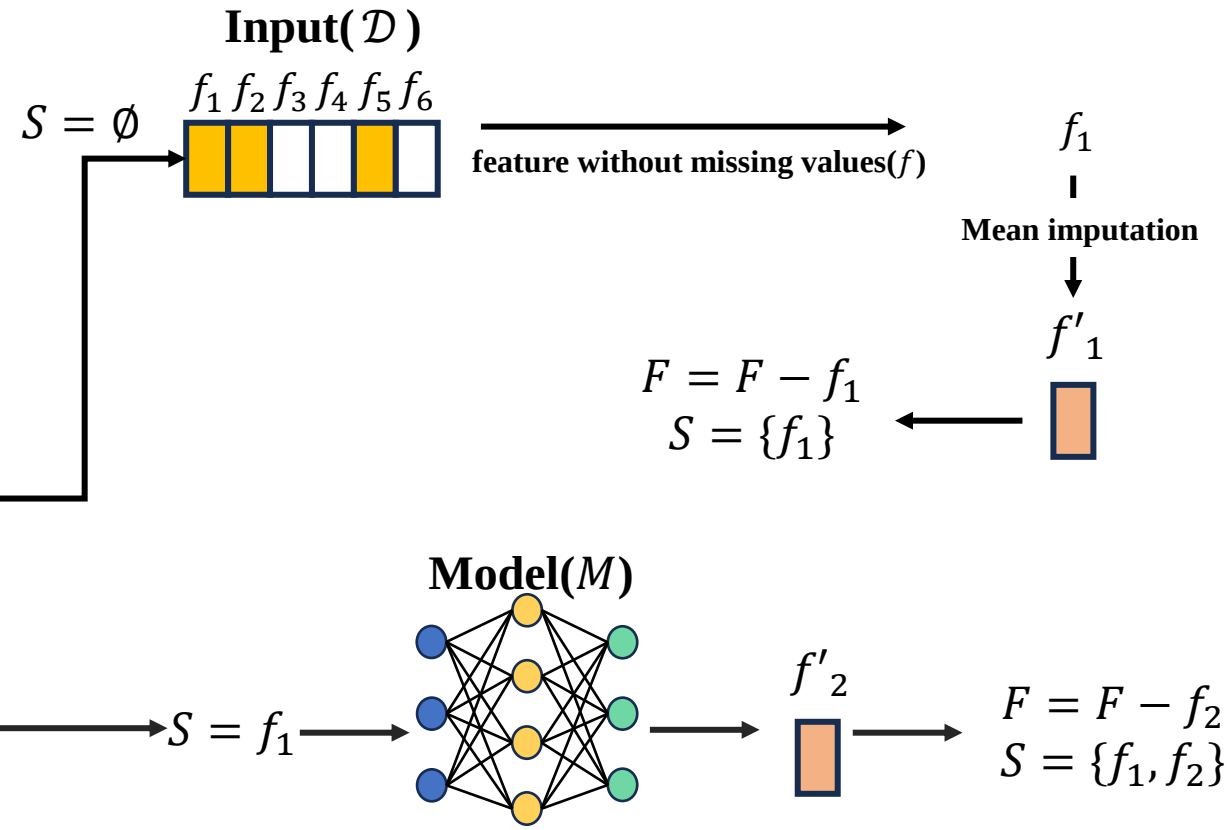
## Pseudo code

- If $S$ is still empty, Choose the first feature $f$. Replace $f$ with its mean using the meanImputer function, denoted as $f'$. Remove $f$ from $F$ and add it to $S$.
- If $S$ is not empty, the method imputes missing values for features in $F$. For each feature $f$ in $F$, it replaces the missing values using the $M$ function. After imputation, it removes $f$ from $F$ and adds it to $S$.

**Algorithm 1** Procedures of proposed imputation method

**Input** : Entire dataset $D$ (with missing values)
**Output:** Imputed dataset $D_{\text{imputed}}$

$S \leftarrow \emptyset$
**for** *each feature* $f$ *in* $D$ **do**
  **if** $f$ *has no missing values* **then**
    $S \leftarrow S\{f\}$
**if** $S = \emptyset$ **then**
  Choose the first feature $f \in F$
  $f' \leftarrow meanImputer(f)$
  $F = F \setminus \{f\}$
  $S \leftarrow S \cup \{f\}$
**for** *each feature* $f$ *in* $F$ **do**
  $f' \leftarrow M(S, f)$
  $F = F \setminus \{f\}$
  $S \leftarrow S \cup \{f\}$



# Experimental Settings

## Dataset

- 25 UCI machine learning repository

| Dataset | Domain | Instances | Attributes | Type |
|---|---|---|---|---|
| Abalone | Biology | 4176 | 9 | Categorical |
| Adult | Social | 32560 | 15 | Categorical |
| Ai4i | Computer Science | 10000 | 6 | Real |
| Bank-marketing | Business | 4521 | 17 | Categorical |
| Blood | Business | 748 | 5 | Real |
| Breast | Health, Medicine | 285 | 10 | Categorical |
| Contraceptive | Health, Medicine | 1473 | 10 | Categorical, Integer |
| Credit | Business | 689 | 16 | Integer, Real |
| Echocardiogram | Heath, Medicine | 129 | 12 | Integer, Real |
| Forty | Social | 320 | 10 | Integer, Real |
| Haberman | Heath, Medicine | 305 | 4 | Integer |
| Heart | Health, Medicine | 302 | 14 | Categorical |
| Hepatitis | Heath, Medicine | 154 | 20 | Integer, Real |
| Iris | Biology | 149 | 5 | Real |
| Liver | Health, Medicine | 344 | 7 | Categorical |
| Lymph | Health, Medicine | 148 | 19 | Categorical |
| Magic | Physics, Chemistry | 19019 | 11 | Real |
| Maternal | Health, Medicine | 1014 | 7 | Real |
| Monk | Social | 415 | 7 | Categorical |
| National-health | Health, Medicine | 2278 | 9 | Real |
| Obesity | Health, Medicine | 2111 | 17 | Integer |
| Rice | Biology | 3809 | 8 | Real |
| Wholesale | Business | 440 | 7 | Integer |
| Wine | Social | 177 | 14 | Integer, Real |
| Yeast | Biology | 1484 | 9 | Real |

## Cross Validation

- Number of Iterations: 30
- Data Split Ratio (Train : Test) = 0.8 : 0.2

# Experimental Settings

## Hyperparameter Settings

| Method | Batch size | Epochs | Loss function | Optimizer | Learning rate |
|---|---|---|---|---|---|
| Proposed | 8 | 10 | Regression | ReLU | 4e-3 |
| DW | | | | | |
| *k*NN | 32 | 50 | Cross-entropy [4] | Adam [4] | 1e-3 [4] |
| SE | | | | | |

## Evaluation Measures

- *Root Mean Squared Error(RMSE)*

→ **Measure the difference between the original data and the predicted data (Lower value, Higher performance).**

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y - \hat{y})^2}{n}}$$

[4] Jongmin Han and Seokho Kang. Dynamic imputation for improved training of neural network with missing values. Expert Systems with Applications, 194(116508):1-8, 2022

# Overall Experimental Results

## Comparison results

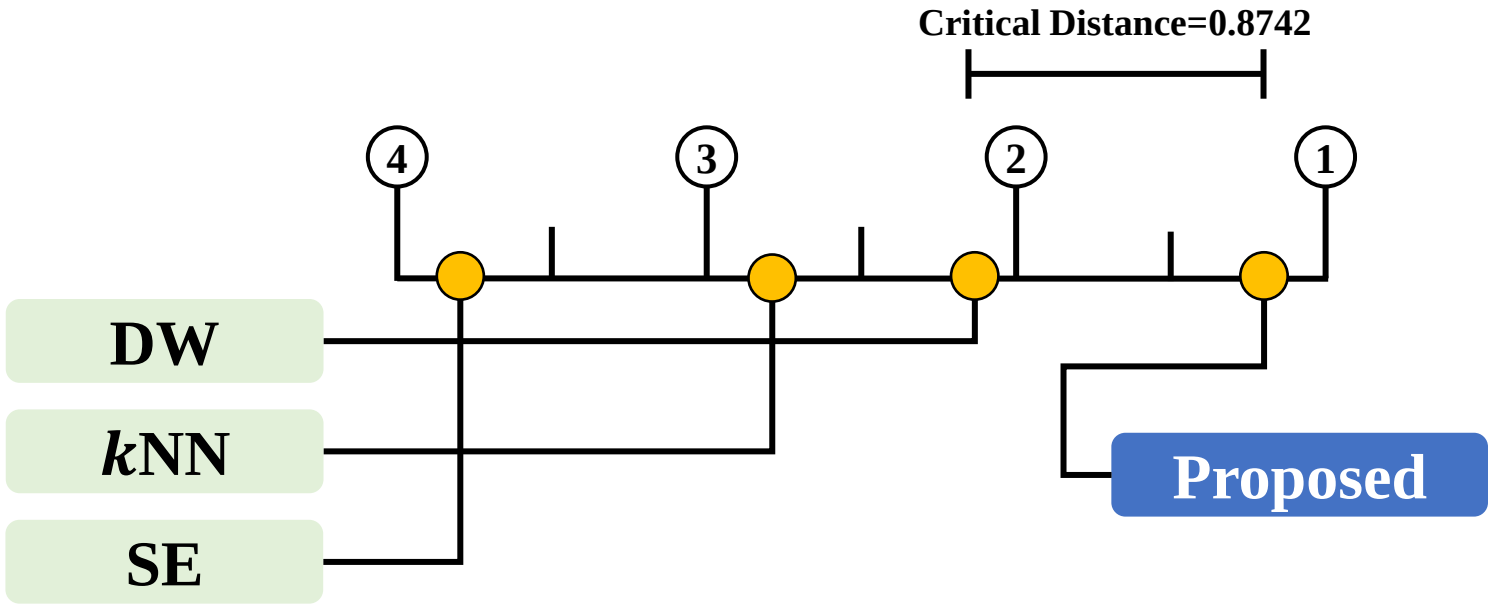
| Dataset | Proposed | DW | *k*NN | SE |
|---|---|---|---|---|
| **Abalone** | **0.2007 ± 0.0599** | 0.5095 ± 0.0100 | 0.5531 ± 0.0103 | 0.6562 ± 0.0025 |
| **Adult** | **0.1894 ± 0.0946** | 0.2146 ± 0.0011 | 0.2582 ± 0.0006 | 0.5803 ± 0.0010 |
| **Ai4i** | **0.1658 ± 0.0401** | 0.1724 ± 0.0009 | 0.2448 ± 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 |

# Overall Experimental Results

## Friedman Test

| Evaluation Measure | Friedman statistic | Critical Values (α = 0.05) |
|---|---|---|
| RMSE | 55.867 | 2.276 |

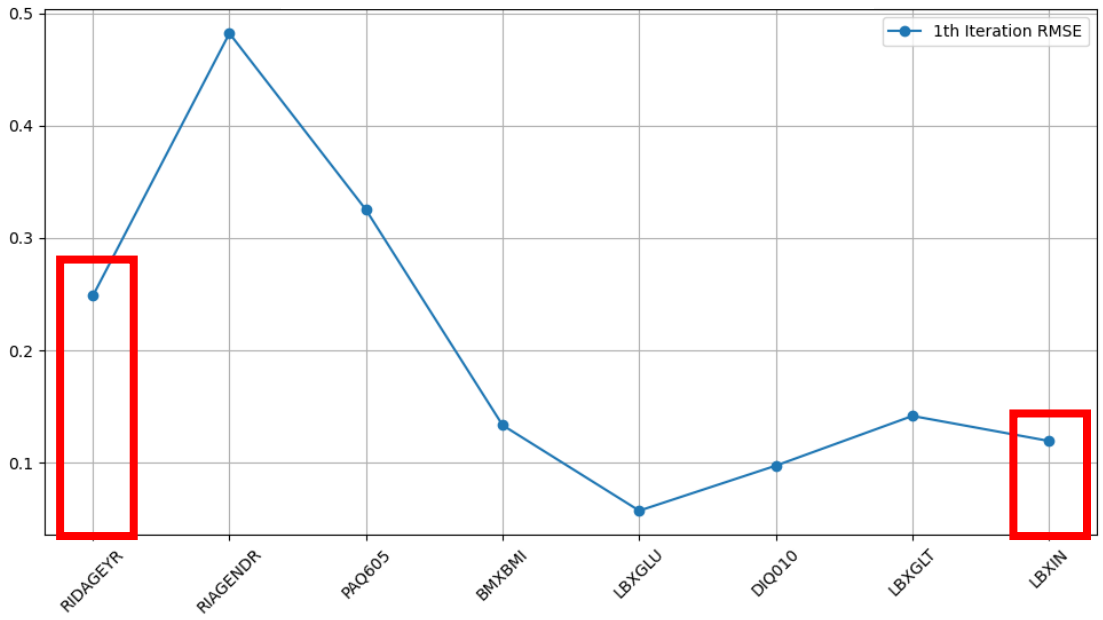
## Bonferroni-Dunn Test



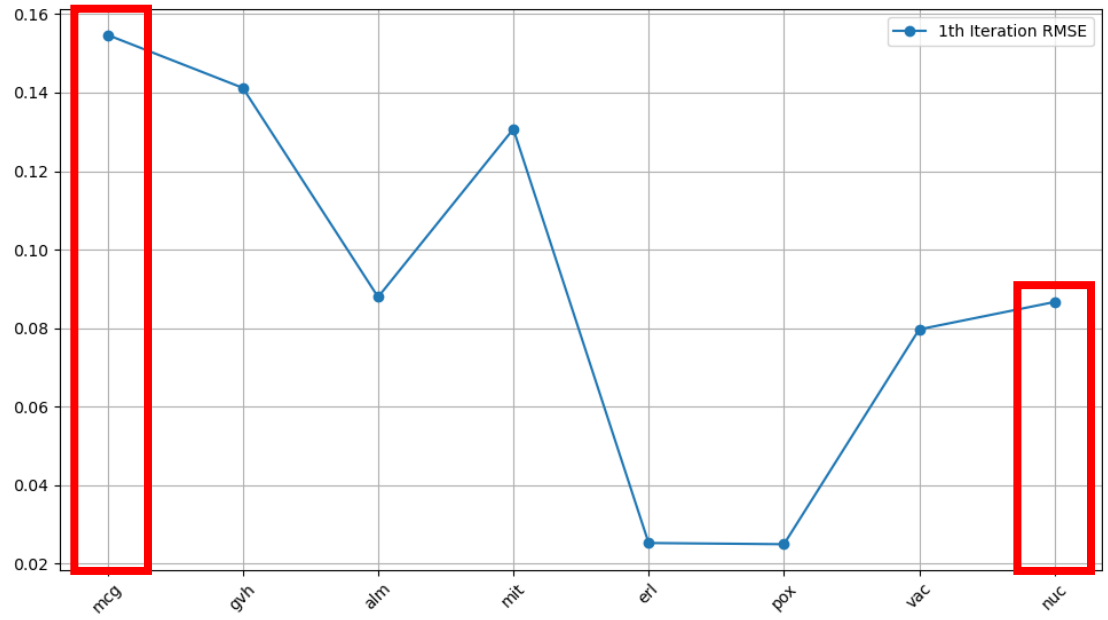
# In-depth Analysis

## In-depth Analysis

- The following graph represents the RMSE performance in feature. It represents two datasets (National-health, Yeast) out of 25. The x-axis represents the features, and the y-axis represents RMSE.
- The graph shows that the performance of the first feature is based on Mean imputation, while the others are based on neural imputation. Neural imputation yields superior performance.




**National**




**Yeast**

Comparison of number of training features

# Conclusion and Contribution

## Conclusion

- MVI research aims to develop and validate methods for accurately replacing missing values and enhancing dataset completeness and reliability for data analysis and modeling.
- Recent interest has focused on employing ANN for MVI to address bias and incomplete datasets, emphasizing the importance of training more features and relationships in the data.
- The proposed imputation method is superior to conventional methods and holds significant promise for real-world applications. Its success is imputed to training on more features, resulting in more accurate and reliable results.

## Contribution

- 2023 Singapore K-Tech Festival - AI/ML Innovation Research Center Booth, Singapore, Singapore, 2023
- 2023 SW Education Festival - CAU Booth, Ilsan, Korea, 2023
- 2023 CAU AI Symposium - AutoML Lab Booth, Seoul, Korea, 2023
- Performance Analysis of Missing Value Imputation for Tabular Data, ICEIC, 2023

# Thank you

# Reference

[1] Biessmann, Felix, et al., DataWig: Missing value imputation for tables. Journal of Machine Learning Research 20.175 (2019): 1-6.

[2] Della Murbarani Prawidya Murti, et al., K-Nearest Neighbor (K-NN) based Missing Data Imputation, 2019 5th International Conference on Science in Information Technology (ICSITech), Yogyakarta, Indonesia, (2019): 83-88.

[3] Turki Aljrees, Improving prediction of cervical cancer using KNN imputer and multi-model ensemble learning. PLOS ONE 19(1) (2024): e0295632.

[4] Jongmin Han and Seokho Kang. Dynamic imputation for improved training of neural network with missing values. Expert Systems with Applications, (2022) 194(116508):1-8.

[5] Janez Demsar, Statistical comparisons of classifiers over multiple data sets. The Journal of Machine learning research 7 (2006): 1-30.