CN-GAIN: Classification and Normalization-Denormalization-Based Generative Adversarial Imputation Network for Missing SMES Data Imputation

Antonius Wahyu Sudrajat[1], Ermatita[2]*, Samsuryadi[3]
Doctoral Program in Engineering Science, Universitas Sriwijaya, Palembang Indonesia[1]
Faculty of Computer Science, Universitas Sriwijaya, Palembang Indonesia[2, 3]
Faculty of Computer Science and Engineering, Universitas Multi Data Palembang, Palembang, Indonesia[1]

***Abstract*—Quality data is crucial for supporting the management and development of SMES carried out by the government. However, the inability of SMES actors to provide complete data often results in incomplete dataset. Missing values present a significant challenge to producing quality data. To address this, missing data imputation methods are essential for improving the accuracy of data analysis. The Generative Adversarial Imputation Network (GAIN) is a machine learning method used for imputing missing data, where data preprocessing plays an important role. This study proposes a new model for missing data imputation called the Classification and Normalization-Denormalization-based Generative Adversarial Imputation Network (CN-GAIN). The study simulates different patterns of missing values, specifically MAR (Missing at Random), MCAR (Missing Completely at Random), and MNAR (Missing Not at Random). For comparison, each missing value pattern is processed using both the CN-GAIN and the base GAIN methods. The results demonstrate that the CN-GAIN model outperforms GAIN in predicting missing values. The CN-GAIN model achieves an accuracy of 0.0801% for the MCAR category and shows a lower error rate (RMSE) of 48.78% for the MNAR category. The mean error (MSE) for the MAR category is 99.60%, while the deviation (MAE) for the MNAR category is 70%.**

***Keywords*—Missing values; GAIN method; normalization-denormalization; imputation; UMKM data**

I. Introduction

Indonesia's SMES (Micro, Small, and Medium Enterprises) are essential in increasing economic growth and regional income. This drives the Indonesian government to continue to develop SMEs through several schemes, including providing business capital, increasing business capacity through training, and so on. As a basis for developing SMES, the government requires SMES characteristic data as a basis for decision-making. Business Intelligence is a technology to support government work in managing SMES data. Data integration is an essential foundation of business intelligence.

Extract Transform Load (ETL) is an essential process in data integration where data processing is carried out. In the data integration process, many problems will affect data quality. One of the challenges in this process is handling missing values. Missing values are problems that arise in the ETL process, more precisely in the data extraction step. [1]. The quality of the underlying data largely determines the quality of the extracted knowledge. Therefore, data quality is a significant concern in data analysis, and data quality is a prerequisite for obtaining quality knowledge. Missing value problems occur due to missing values from an attribute caused by errors when collecting data, system errors ([2], [3], errors in data entry, refusal or inability of respondents to provide accurate answers [4] and merging of unrelated data [5]. Missing value is a fundamental problem in data science [6].

In some applications, missing values cannot be tolerated and must be replaced with concrete values [7]. Related studies have shown that missing value imputation is beneficial and is a better option than data deletion [8]. Missing data imputation means replacing or correcting the missing data with reasonable values to achieve completeness [9]. Missing data imputation is essential because decision-making errors will occur when an incomplete data set is supported [10]. Some important impacts of handling missing data include the accuracy of statistical analysis, better interpretation, reduction of bias, and improvement of data quality ([3], [11]).

The missing value imputation approach can be broadly categorized into traditional methods and Machine Learning (ML) based algorithm methods. Traditional methods include mean [12], median, linear regression [13], and mode. Some ML-based methods include Algorithms Clustering[14], K-Narest Neighbor (KNN) [15], Support Vector Machine (SVM) [16], Decision Trees (DT) [17], [18], Random Forest (RF)[19] dan Generative Adversarial Networks (GAN) ([20], [21], [22], [23], [24]). The ability to optimize and extract relationships between data points is an advantage of machine learning-based methods [7]. GAN is an ML method that has attracted researchers' attention in recent years. Missing values are a significant problem in data mining, big data analysis, and ML-based decision-making flows, as the final mining or analysis results can be adversely affected when incomplete data is not imputed correctly [25]. Improvement efforts have been made in several studies that underlie the GAN method, including the research presented in [26] proposes improvements in a new method, namely Generative Adversarial Imputation Nets (GAIN) [27]. In this method, the generator accurately imputes

missing data, and the discriminator aims to distinguish between observed and imputed components. Further improvements to GAIN are carried out by research [7], where the idea is to use deconvolution on the generator and discriminator (DEGAIN). This method makes improvements by adding deconvolution to eliminate correlations between data. Improvements to the imputation method are made based on the characteristics of the data structure ([28], [29] and the characteristics of the data values. At the same time, research that focuses on the characteristics of data values is still rarely done. The characteristics of the data values are an important initial step to perform accurate imputation. High differences in data values will result in inaccurate results in data processing.

In this study, we optimized the GAIN method [27], a GAN-based algorithm, by developing an enhanced version referred to as CN-GAIN. The CN-GAIN method improves upon GAIN by incorporating data preprocessing tasks as an initial step before imputation, taking into account the characteristics of the existing data. These preprocessing steps include data classification using the k-means method and normalization/denormalization using a robust scaler. The purpose of data classification is to categorize the data based on its inherent characteristics [30]. Meanwhile, normalization and denormalization ensure that no data values disproportionately dominate the dataset. We evaluated the performance of our proposed method using a dataset of SMES from a district in South Sumatra Province. The evaluation included measuring accuracy and several error metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Squared Error (MSE). We compared the performance of CN-GAIN with the standard GAIN algorithm.

This study proposes a new method for handling missing values, with the basic method being GAIN. This paper is structured as follows. Section II discusses related works in handling missing data, especially those based on the GAIN method and the improvement efforts made. Section III explains the efforts made by researchers in handling missing values through a series of stages and the use of methods so that they can produce more accurate imputation values. Section IV carries out each planned stage, which is very important in research to determine the proposed method's results. Section V discusses the results of each stage and the results obtained. Section VI presents the conclusions of the research and future research plans.

II. Literature Review

Missing value is a widespread problem encountered in many data collection cases. The missing value is a value that is not stored for a variable in the desired observation [30]. Missing data is grouped into three categories, namely: (1) Missing Completely at Random (MCAR), where the data is lost entirely at random (no dependence on any variable). (2) Missing at Random (MAR), where the missing data depends on the observed variables. (3) Missing Not at Random (MNAR), where the missing data depends on the observed variables and unobserved variables [31]. Data imputation is a common way to deal with missing values where missing values are replaced by applying various methods [32]. Missing value imputation is a valuable solution in cleaning datasets, and generative machine learning methods can produce data that is completely indistinguishable from reality. Research in missing data imputation has mainly focused on adapting existing methods to suit specific datasets and operational environments, improving model adaptability and accuracy.

Generative Adversarial Net (GAN) is an artificial intelligence algorithm designed to solve generative modeling problems. GAN algorithms can be used to fill in gaps in missing data [33]. Many attempts have been made to improve GAN-based missing data imputation ([34], [35], [36][37]). One of the GAN-based imputation methods is Generative Adversarial Imputation Nets (GAIN). In GAIN, the generator component (G) takes a real data vector, imputes missing values conditioned on the actually observed data, and gives a complete vector. Then, the discriminator component (D) gets the complete vector and tries to determine which elements are actually observed and which are synthesized [31].

Several researchers have made improvements to the GAIN method. For example, [38] focused on repairing missing data in single-cell datasets using the basic GAIN method. The proposed method is Single-Cell Generative Adversarial Imputation Nets (scGAIN). Furthermore, the research conducted [39] proposed the Generative Adversarial Multiple Imputation Network (GAMIN) method for duplicate data imputation with data loss rate more than 80%. This method is applied to image data. Optimization of the GAIN method is also carried out by [40] by performing a pre-training procedure to learn the potential information contained in the data and classifying the data using synthetic pseudo-labels which are then named Pseudo-label Conditional Generative Adversarial Imputation Networks (PC-GAIN).

In research conducted by [41] proposed the Deconvolutional Generative Adversarial Imputation Network (DEGAIN) method, which makes improvements by adding deconvolution to eliminate correlations between data. The research [42] proposed a new missing data imputation model based on data clustering with the basic GAIN method as input data. The data set used is electricity consumption with the MCAR missing value type. This imputation method is then named Clustering and Classification-based Generative Adversarial Imputation Network (CC-GAIN). CC-GAIN aims to enhance imputation accuracy by considering both time-series and pattern features in building electricity consumption data.

TABLE I. Previous Research

Ref.	Year	Method	Dataset	Type Data
[38]	2019	scGAIN	Tow dataset: Simulated and rela-word dataset	Numeric
[39]	2020	GAMIN	MNIST and CelebA	Image
[40]	2021	PC-GAIN	UCI repository and MNIST dataset	Numerical, categorical and image
[41]	2023	DEGAIN	Letter and SPAM	Image
[42]	2024	CC-GAIN	Electricity consumtion data	Numeric

Table I summarizes previous research and is the basis for this research. Based on the research that has been described

previously, no previous research has implemented data preprocessing steps such as data classification, normalization, and denormalization before imputing missing data with the GAIN method. Incorporating these preprocessing steps can better prepare the data for the imputation process and potentially improve the overall performance of the method.

III. Research Methodology

Data completeness is one of several dimensions measured in determining data quality. As a data quality dimension, data completeness means the data set is free from missing values (MV or NA). Fig. 1 shows the flow carried out in this study. In this study, the steps taken are collecting data sets, creating data sets (simulation), and applying data sets to the proposed method, namely CN-GAIN and its basic method, GAIN. The last is to evaluate the data set through the prediction process.

A. Data Set

The data set used in this study is the SMES data set in South Sumatra, which was collected from 2017-2020 by the Dinas Koperasi dan UKM South Sumatera. This data is collected per period using several mechanisms, including through distributed data sheets or direct data collection by officers. There are 3301 SMES data records that were successfully collected. The fields include the type of business, manpower, investment_value, production_capacity, production_value, and bb_bp_value.

The performance of the proposed CN-GAIN imputation model is tested using SMES data as described in Table II with predetermined attributes. Table III shows the characteristics of the SMES dataset.

TABLE II. Data Set UMKM

Type of business	manpower	Investment_value	Production_capacity	Production_value	bb_bp_value
Tempe	3	5000	75000	6000	20000
Tempe	3	5000	30000	30000	10000
Tempe	6	5000	75000	75000	25000
Tahu	1	5000	714000	285600	95200
Tahu	2	5000	36000	108000	36000
Tahu	2	2500	48000	14400	4800
Tahu	1	1500	90000	45000	15000
Batu bata	1	800	300000	165000	55000
Batako	3	85000	300000	750000	25000
Meuble	3	3000	1920	42350	14416
Meuble	3	10000	888	58000	19333
Meuble	3	15000	1800	52500	17500
Bengkel	3	10000	960	240000	8000
…	…	…	…	…	…

TABLE III. Characteristics of UMKM Dataset

Field	Data type	Description
Type of business	String	Types of SMES businesses
Manpower	Integer	Number of workers
Investment_value	Integer	Business investment value
Production_capacity	Integer	Production capacity
Production_value	Integer	Production value
bb_bp_value	Integer	Raw material value

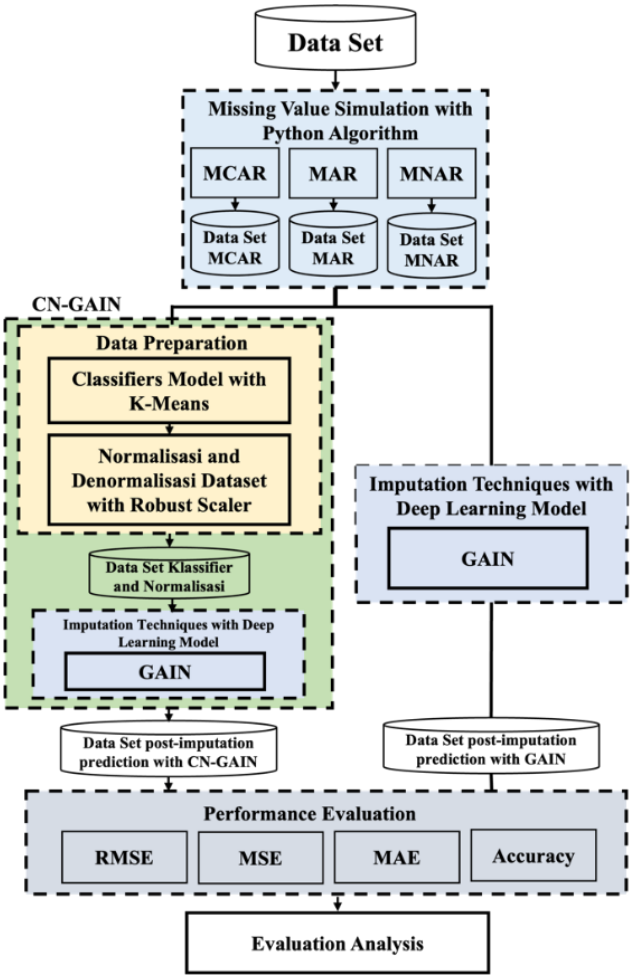


Fig. 1. Experimental flowchart.

B. Missing Value Simulation

Based on the dataset that has been obtained, the next step is to simulate the missing data. In this experiment using Python, the data set was randomly simulated into the MCAR, MNAR, and MAR categories. Algorithm 1 is a pseudo-code of the missing value simulation. By using this algorithm, complete SMES data is then removed randomly. Several Python libraries are used in this simulation, namely wget and numpy.

Algorithm 1. Pseudo-code of Missing Values simulation

```
import numpy as np
import pandas as pd
from utils import *
import torch
Function produce_NA(X, p_miss, mecha="MCAR",
opt=None, p_obs=None, q=None):
    If mecha is "MAR":
    mask = Generate MAR
Else if mecha is "MNAR":
    mask = Generate MNAR
Else:
    mask = Generate MCAR
Return value
End Function
```

This function generates missing values in a dataset. The function relies on several libraries: numpy for numerical computations, pandas for data manipulation, and torch for deep learning tasks. The function takes the following parameters: X for the input dataset, p_miss for the proportion missing values, mecha for the mechanism for generating missing data, which can be MCAR (Missing Completely At Random), MAR (Missing At Random), or MNAR (Missing Not At Random), opt, p_obs, q are optional parameters. The function would return the dataset (X) modified with the generated missing values according to the selected mechanism.

C. Data Preparation

The data preparation stage is carried out to understand the characteristics of the data. This step consists of a collection of techniques applied to the data to improve the data quality before processing the machine learning data. Where the initial step taken is to carry out the Classifier model with K-Means, perform data normalization and data denormalization.

1) Classifier model with k-means: The k-means algorithm is the simplest and most commonly used clustering algorithm. This algorithm determines the number of clusters (k) that need to be grouped in a Data Set. The steps in performing clustering with the K-Means method include the following [43]:

a) Determine the number of centroids

b) Determine points or centroids randomly

$$D(x,y) = \sqrt{(X_1 - Y_1)^2} + (X_2 - Y_2)^2 \quad (1)$$

c) Calculate and assign new centroids for each cluster.

$$c = \frac{\sum m}{n} \quad (2)$$

d) Repeat step c, until there are no further changes.

2) Normalization and denormalization dataset with robust scaler: Normalization is done to change the value of the attribute in the dataset to have a uniform scale or range. Normalization is used to improve accuracy in classification. [44]. In this study, the normalization technique used is robust scaler. The Robust Scaler formula is stated in Eq. (3):

$$X_{scaled} = \frac{X - median(X)}{IQR(X)} \quad (3)$$

Denormalization using the Robust Scaler involves reversing the scaling process to convert the scaled data back to its original values. This is particularly useful when you want to interpret the results of your model in the context of the original data.

$$X_{Original} = \left(X_{scaled} \times IQR(X)\right) + Media(X) \quad (4)$$

D. Imputation Techniques with GAIN Model

Generative Adversarial Network (GAN) is an ML framework trained with two neural networks, namely the generator and the discriminator. The generator aims to create synthetic data that resembles real data, while the discriminator aims to distinguish between real and generated samples [46]. The GAN method is designed to generate images, GANs have been applied in various fields, including natural language processing and speech processing. The goal of GAN is to

generate images that are very similar to real images. GANs are designed for adversarial training of the generator (G) and the discriminator (D), where G is trained to create data that is most similar to the real data, and D is trained to classify the data generated by G. In the process of imputing missing data, GANs generate values that are similar to the real values by modeling the distribution of data surrounding the missing values.

1) The generator: The generator is trained for missing data imputation. In the generator model G, the input value is X_a and the matrix M and the noise variable Z are obtained.

$$X_m = X_a \odot M + Z \odot (1 - M) \quad (5)$$

$$X_f = X_a \odot M + G(X_m) \odot (1 - M) \quad (6)$$

2) The discriminator: D is used to distinguish the imputed data through G. Unlike GAN, it distinguishes between true and false data from certain constituent elements, not all generated data.

3) Hint generator: The hint (H) generator provides some information on the mask M to guide the training of D. This can prevent G and D from learning unintended distributions:

$$H = B \odot M + 0.5 \odot (1 - B) \quad (7)$$

E. Performance Evaluation

This phase is used to evaluate the performance of the proposed method. In this study, accuracy measurement was conducted, and three error metrics were employed to assess performance: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Squared Error (MSE). The rationale for using multiple error metrics is to comprehensively compare errors, as each metric offers different advantages, disadvantages, and features. The mathematical formula for calculating accuracy is presented in Eq. (8):

$$Accuracy = \frac{TN+TP}{TN+FN+FP+TP} \quad (8)$$

Root Mean Square Error (RMSE) calculates the error between the real value and the estimated value (imputed value) to measure the accuracy of imputation ([45], [46]). The difference between the predicted (hypothetical) value and the actual value. RSME is mathematically expressed as shown in Eq. (9):

$$RSME = \sqrt{\frac{\sum_{i=1}^{n}(X_i^{a}bs - X_i^{i}mputed)^2}{n}} \quad (9)$$

Mean Absolute Error (MAE) is a matrix for calculating positive and negative deviations between $\hat{y}_i$ predicted and actual values ([46], [47]). MAE is mathematically expressed as shown in Eq. (10):

$$MAE(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (10)$$

Mean Squared Error (MSE) calculates the average error. The average value is close to zero but not negative [46].

Mathematically, MSE is defined based on Eq. (11):

$$MSE(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (11)$$

IV. Experimental Results

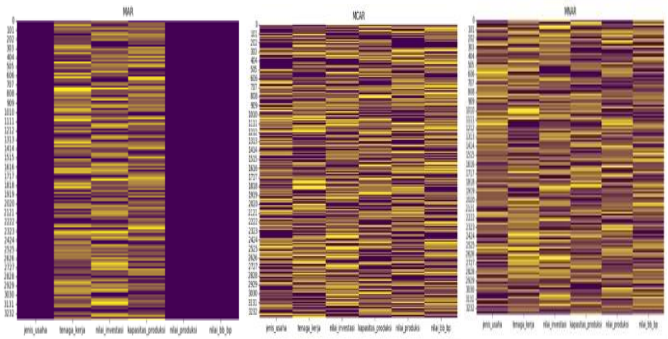
This section presents the results of each step that has been explained previously, namely the results of the missing value simulation, data pre-processing and the proposed CN-GAIN algorithm model.

A. Missing Values Simulation

Based on the python algorithm that has been created previously, where the data set is categorized into three categories of missing values, namely: MAR, MCAR and MNAR. Table IV is the result of the missing value algorithm process that has been run. While Fig. 2 is a visualization of the percentage of missing values based on the missing value category.

TABLE IV. Percentage of Missing Values in Each Attribute

No	Field	MAR		MCAR		MNAR	
		MV	%	MV	%	MV	%
1	Type of business	0	0	1293	39.17	1304	39.50
2	Manpower	1353	40.99	1312	39.75	1349	40.87
3	Investment_value	1351	40.93	1326	40.17	1350	40.90
4	Production_capacity	1334	40.41	1321	40.02	1317	39.90
5	Production_value	0	0	1309	39.65	1327	40.20
6	bb_bp_value	0	0	1338	40.53	1348	40.87



(a) MAR (b) MCAR (c) MNAR

Fig. 2. Missing value visualization.

B. Classificaton with K-Means

Classification is carried out on datasets based on business type. jenis_usaha column appears to contain information about the type of business, but there are many unique values with variations in capitalization and wording (e.g., "Bengkel motor" and "Bengkel Motor"). Additionally, some values are particular (e.g., "Bengkel"). Hence, classification is needed. To classify these values, K-means is employed. From the existing SMES data, 10 classifications were obtained, as shown in Table V.

TABLE V. CLASSIFICATION OF UMKM DATA

Cluster	Businesses
0	Crafts
1	Vehicle Repair and Maintenance
2	Brick-making
3	Furniture, Woodworking
4	Fish Products
5	Other product manufacturing
6	Tempeh and Tofu Production
7	Agriculture and Farming
8	Snack Production
9	Beverage

C. Architecture of the Proposed CN-GAIN Model

The proposed CN-GAIN model for UMKM data imputation is based on the GAIN method. This model consists of five main modules, namely: 1) clustering, 2) normalization, 3) denormalization, 4) generator, 5) discriminator.

Algorithm 1. Pseudo-code of CN-GAIN

```
from sklearn.preprocessing import RobustScaler
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow.keras.layers import Input, Dense
from tensorflow.keras.models import Model
Determine the classification
Draw dataset, number of clusters k
   For i = 1, …, do
      Label (i) ← clastering(k)
   end for
Normalisasi
scaler = Initialize RobustScaler()
data_scaled = Fit the scaler to the dataframe (df) and transform the
data
Denormalization
median_values = Retrieve the median of each feature from the scaler
iqr_values = Retrieve the interquartile range (IQR) of each feature
from the scaler
   For each feature in data_scaled:
         original_data = (data_scaled * iqr_values)+
         median_values
Update Generator and Discriminator
Discriminator optimization
While training loss has not converged do
Draw samples from the dataset
For j = 1, …,samples do
X_m(j) ← G(X_a(j), z(j))
X_f(j) ← m(j)⊙(X_a(j) + (1 − m(j))⊙X_m(j)
h(j) ← b(j)⊙m(j) + 0.5(1 − b(j))
y(j) ← C(X_f(j), l(j))
end for
update D using adam optimizer
Generator Optimization
Draw samples from the dataset
For j = 1, …, do
h(j) ← b(j)⊙m(j) + 0.5(1 − b(j))
y(j) ← C(X_f(j), l(j))
end for
update D using adam optimizer
```

Data was normalized using a Robust Scaler by subtracting the median and dividing the Inter Quartile Range (IQR). Then, data was deformalized using Robust Scaler. The optimization begins with discriminator D, which is tuned using a fixed generator and classification via mini-batch sD. Independent samples of Z and B, represented as z(j) and b(j), generate h(j) and Xf(j), respectively. Additionally, y(j) is derived from Xf(j) and l(j). The discriminator is optimized using Xf(j), h(j), and y(j) across all mini-batches. Then, generator G will be optimized by mini-batch s(G) while the discriminator D is updated, and the classification C remains fixed. h(j) and y(j) are computed for all mini-batches and used in optimizing G.

V. RESULT AND DISCUSSION

The dataset prepared and simulated into missing data categories (MAR, MCAR, and MNAR) is then used with the proposed steps and techniques in the data processing stage.

A. Result

The performance of the missing value imputation of the proposed CN-GAIN model is compared with the baseline model, namely the GAIN Model. Each model was tested on three missing value data categories: MAR, MCAR, and MNAR. Where each category has a different level of missing value, in this study, the percentage of missing values was created randomly using the library in python for each type of missing value category. This is different from what was done in the study [42], where the percentage of missing values ranges from 10% to 90%.

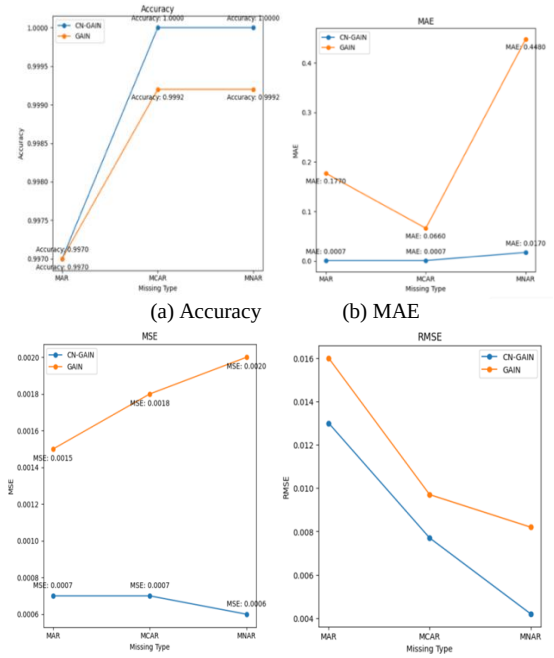
The proposed CN-GAIN model has a better accuracy rate and a low error rate. An important step is applying classification, normalization, and denormalization of data before the imputation process with the GAIN model. As a comparison, researchers also applied the GAIN model to compare accuracy and error rates. Table VI shows the results of the proposed model trial with the base model after imputation.

TABLE VI. PERFORMANCE EVALUATION OF PROPOSED CN-GAIN AND GAIN

Matrics	MAR		MCAR		MNAR	
	CN-GAIN	GAIN	CN-GAIN	GAIN	CN-GAIN	GAIN
Acuracy	0.997	0.997	1.00	0.9992	1.00	0.9992
RSME	0.013	0.016	0.0077	0.0097	0.0042	0.0082
MSE	0.0007	0.0015	0.0007	0.0018	0.0006	0.0020
MAE	0.0007	0.177	0.0007	0.066	0.017	0.448

Based on the results of the trials conducted for accuracy, the MAR missing value category has the same value, which is 0.9970. for the MCAR category, the CN-GAIN method has a better accuracy level of 1.00 while the GAIN method is 0.9992. as well as for the value in the MNAR missing value category. For performance seen from the Root Mean Square Error (RMSE) has a better value, whereas the MNAR missing value category has a better value, which is 0.0042. While the performance for the Mean Absolute Error (MAE) shows a better value than this method is MAR, which is 0.0007. Finally, the performance of the Mean Squared Error (MSE) category of MNAR missing value has the best value of the proposed model, which is 0.0006. Fig. 3 compares the CN-GAIN and the GAIN methods as the basic methods for three types of missing values: MAR, MCAR, and MNAR.

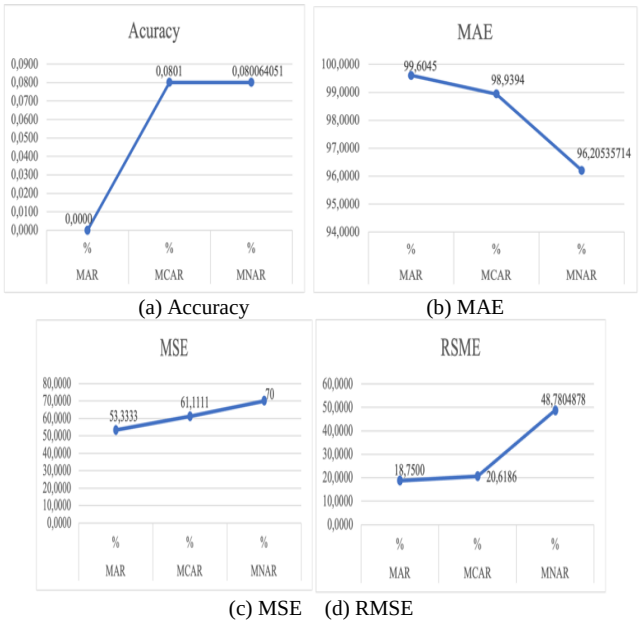
Evidence of the proposed model's improvement is demonstrated by the increased performance percentage of the CN-GAIN model compared to the base model, GAIN. Fig. 4 is a visualization of the presentation of CN-GAIN performance on each type of missing value compared to the basic GAIN model. The imputation value's accuracy level occurred in the MCAR missing value category type of 0.0801% and MNAR of 0.0800, while for the MAR category type there was no increase. For the percentage of actual and imputed values (RMSE), the MNAR missing value category type has a better improvement of 48.780%.



(a) Accuracy (b) MAE

(c) MSE (d) RMSE

Fig. 3. Performance evaluation of different methods.



(a) Accuracy (b) MAE

(c) MSE (d) RMSE

Fig. 4. Performance percentage of CN-GAIN compared to GAIN method on missing value type.

B. Discussion

Based on the results of the trials conducted, the CN-GAIN model has a better level of accuracy than its basic model, the GAIN model. While for the error rate the CN-GAIN method also has a lower error rate. The results (Table V) show that of the three types of missing values simulated, the CN-GAIN model has a relatively high increase in accuracy; only the MAR type of missing value has the same value when the accuracy level is measured. These results prove that the CN-GAIN model as a framework for handling missing values is proposed to be implemented. The classification steps carried out in data preparation make the data set classified according to its group and normalization-denormalization makes the data set in a value range that is not too far apart. This step improved the imputation process, even though there was no change or significant improvement in one type of missing value, especially in terms of accuracy in the MAR type of missing value.

From the overall results, this study has proven that the framework developed using the proposed classification and normalization-denormalization has a better level of accuracy than its standard. This study also proves that the data preprocessing carried out has a better impact on the quality of the data obtained after the value prediction is carried out.

VI. Conclusion and Future Work

Incomplete SMES data is a significant challenge for SMES' proper management and development. Effective data imputation is essential to produce quality SMES data. In this study, we propose CN-GAIN, a new missing data imputation method designed to handle data with multiple data characteristics in SMES data. By making efforts to classify, normalize, and denormalize data in the data pre-processing process before imputation using the GAIN method. The CN-GAIN model performs better in predicting missing values, with an accuracy value of 0.0801% for the MCAR category and a lower error rate (RMSE), of 48.78% for the MNAR category. The average error (MSE) is 99.60% for the MAR category, and the deviation value (MAE) is 70% for the MNAR category.

For further research, researchers will test the model on other data sources with more complex data characteristics with more varied data types.

References

[1] M. Souibgui, F. Atigui, S. Zammali, S. Cherfi, and S. Ben Yahia, "Data quality in ETL process: A preliminary study," Procedia Comput Sci, vol. 159, pp. 676–687, 2019, doi: 10.1016/j.procs.2019.09.223.

[2] M. P. Fernando, F. Cèsar, N. David, and H. O. José, "Missing the missing values: The ugly duckling of fairness in machine learning," International Journal of Intelligent Systems, vol. 36, no. 7, pp. 3217–3258, Jul. 2021, doi: 10.1002/int.22415.

[3] D. Li, H. Zhang, T. Li, A. Bouras, X. Yu, and T. Wang, "Hybrid Missing Value Imputation Algorithms Using Fuzzy C-Means and Vaguely Quantified Rough Set," IEEE Transactions on Fuzzy Systems, vol. 30, no. 5, pp. 1396–1408, May 2022, doi: 10.1109/TFUZZ.2021.3058643.

[4] G. Doquire and M. Verleysen, "Feature selection with missing data using mutual information estimators," Neurocomputing, vol. 90, pp. 3–11, Aug. 2012, doi: 10.1016/j.neucom.2012.02.031.

[5] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona, "A survey on missing data in machine learning," J Big Data, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00516-9.

[6] Z. Chen, S. Tan, U. Chajewska, C. Rudin, and R. Caruana, "Missing Values and Imputation in Healthcare Data: Can Interpretable Machine Learning Help?," 2023.

[7] R. Shahbazian and I. Trubitsyna, "DEGAIN as tool for Missing Data Imputation," 2023. [Online]. Available: http://ceur-ws.org

[8] M. W. Huang, W. C. Lin, C. W. Chen, S. W. Ke, C. F. Tsai, and W. Eberle, "Data preprocessing issues for incomplete medical datasets," Expert Syst, vol. 33, no. 5, pp. 432–438, Oct. 2016, doi: 10.1111/exsy.12155.

[9] T. Thomas and E. Rajabi, "A systematic review of machine learning-based missing value imputation techniques," Data Technologies and Applications, vol. 55, no. 4, pp. 558–585, 2021, doi: 10.1108/DTA-12-2020-0298.

[10] A. R. Ismail, N. Z. Abidin, and M. K. Maen, "Systematic Review on Missing Data Imputation Techniques with Machine Learning Algorithms for Healthcare," Mar. 01, 2022, Department of Electrical Engineering, Universitas Muhammadiyah Yogyakarta. doi: 10.18196/jrc.v3i2.13133.

[11] I. Setiawan, R. Gernowo, and B. Warsito, "A Systematic Literature Review on Missing Values: Research Trends, Datasets, Methods and Frameworks," in E3S Web of Conferences, EDP Sciences, Nov. 2023. doi: 10.1051/e3sconf/202344802020.

[12] F. Yulian Pamuji, Ahmad Rofiqul Muslikh, Rizza Muhammad Arief, and Delviana Muti, "Komparasi Metode Mean dan KNN Imputation dalam Mengatasi Missing Value pada Dataset Kecil," Jurnal Informatika Polinema, vol. 10, no. 2, pp. 257–264, Feb. 2024, doi: 10.33795/jip.v10i2.5031.

[13] N. Karmitsa, S. Taheri, A. Bagirov, and P. Makinen, "Missing Value Imputation via Clusterwise Linear Regression," IEEE Trans Knowl Data Eng, vol. 34, no. 4, pp. 1889–1901, Apr. 2022, doi: 10.1109/TKDE.2020.3001694.

[14] A. Dubey and A. Rasool, "Clustering-Based Hybrid Approach for Multivariate Missing Data Imputation," 2020. [Online]. Available: www.ijacsa.thesai.org

[15] W. Sudrajat and I. Cholid, "K-NEAREST NEIGHBOR (K-NN) UNTUK PENANGANAN MISSING VALUE PADA DATA UMKM," 2023.

[16] A. Syarif, O. Desti Riana, D. Asiah Shofiana, and A. Junaidi, "A Comprehensive Comparative Study of Machine Learning Methods for Chronic Kidney Disease Classification: Decision Tree, Support Vector Machine, and Naive Bayes." [Online]. Available: www.ijacsa.thesai.org

[17] S. Nikfalazar, C. H. Yeh, S. Bedingfield, and H. A. Khorshidi, "Missing data imputation using decision trees and fuzzy clustering with iterative learning," Knowl Inf Syst, vol. 62, no. 6, pp. 2419–2437, Jun. 2020, doi: 10.1007/s10115-019-01427-1.

[18] A. Syarif, O. Desti Riana, D. Asiah Shofiana, and A. Junaidi, "A Comprehensive Comparative Study of Machine Learning Methods for Chronic Kidney Disease Classification: Decision Tree, Support Vector Machine, and Naive Bayes." [Online]. Available: www.ijacsa.thesai.org

[19] A. R. Alsaber, J. Pan, and A. Al-Hurban, "Handling complex missing data using random forest approach for an air quality monitoring dataset: A case study of kuwait environmental data (2012 to 2018)," Int J Environ Res Public Health, vol. 18, no. 3, pp. 1–26, 2021, doi: 10.3390/ijerph18031333.

[20] "Mixed Data Imputation using Generative Adversarial Networks".

[21] H. Ou, Y. Yao, and Y. He, "Missing Data Imputation Method Combining Random Forest and Generative Adversarial Imputation Network," Sensors, vol. 24, no. 4, Feb. 2024, doi: 10.3390/s24041112.

[22] R. Shahbazian and S. Greco, "Generative Adversarial Networks Assist Missing Data Imputation: A Comprehensive Survey and Evaluation," IEEE Access, vol. 11, pp. 88908–88928, 2023, doi: 10.1109/ACCESS.2023.3306721.

[23] W. Dong et al., "Generative adversarial networks for imputing missing data for big data clinical research," BMC Med Res Methodol, vol. 21, no. 1, Dec. 2021, doi: 10.1186/s12874-021-01272-3.

[24] J. Gao, Z. Cai, W. Sun, and Y. Jiao, "A Novel Method for Imputing Missing Values in Ship Static Data Based on Generative Adversarial Networks," J Mar Sci Eng, vol. 11, no. 4, Apr. 2023, doi: 10.3390/jmse11040806.

[25] M. K. Hasan, M. A. Alam, S. Roy, A. Dutta, M. T. Jawad, and S. Das, "Missing value imputation affects the performance of machine learning: A review and analysis of the literature (2010–2021)," Jan. 01, 2021, Elsevier Ltd. doi: 10.1016/j.imu.2021.100799.

[26] J. Yoon, J. Jordon, and M. van der Schaar, "GAIN: Missing Data Imputation using Generative Adversarial Nets," Jun. 2018, [Online]. Available: http://arxiv.org/abs/1806.02920

[27] J. Yoon, J. Jordon, and M. van der Schaar, "GAIN: Missing Data Imputation using Generative Adversarial Nets," Jun. 2018, [Online]. Available: http://arxiv.org/abs/1806.02920

[28] A. M. Sefidian and N. Daneshpour, "Missing value imputation using a novel grey based fuzzy c-means, mutual information based feature selection, and regression model," Expert Syst Appl, vol. 115, pp. 68–94, Jan. 2019, doi: 10.1016/j.eswa.2018.07.057.

[29] H. Rosado-Galindo and S. Dávila-Padilla, "Tree-Based Missing Value Imputation Using Feature Selection," Journal of Data Science, vol. 18, no. 4, pp. 606–631, Oct. 2020, doi: 10.6339/JDS.202010_18(4).0002.

[30] M. El-Bakry, A. El-Kilany, S. Mazen, and F. Ali, "Fuzzy based Techniques for Handling Missing Values." [Online]. Available: www.ijacsa.thesai.org

[31] J. Yoon, J. Jordon, and M. Van Der Schaar, "GAIN: Missing Data Imputation using Generative Adversarial Nets," 2018.

[32] Z. A. Nadzurah, I. Amelia Ritahani, and A. Nurul, "Performance Analysis of Machine Learning Algorithms for Missing Value Imputation," International Journal of Advanced Computer Science and Applications, vol. 9, no. 6, 2018.

[33] I. Goodfellow et al., "Generative adversarial networks," Commun ACM, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: 10.1145/3422622.

[34] D. Lee, J. Kim, W.-J. Moon, and J. C. Ye, "CollaGAN: Collaborative GAN for Missing Image Data Imputation."

[35] S. Wang, W. Li, S. Hou, J. Guan, and J. Yao, "STA-GAN: A Spatio-Temporal Attention Generative Adversarial Network for Missing Value Imputation in Satellite Data," Remote Sens (Basel), vol. 15, no. 1, Jan. 2023, doi: 10.3390/rs15010088.

[36] X. Zheng, Y. Wu, Y. Pan, W. Lin, L. Ma, and J. Zhao, "DPGAN: A Dual-Path Generative Adversarial Network for Missing Data Imputation in Graphs." [Online]. Available: https://github.com/momoxia/DPGAN.

[37] W. Qiu, Y. Huang, and Q. Li, "IFGAN: Missing Value Imputation using Feature-specific Generative Adversarial Networks," in Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 4715–4723. doi: 10.1109/BigData50022.2020.9378240.

[38] M. K. Gunady, J. Kancherla, H. Corrada Bravo, and S. Feizi, "scGAIN: Single Cell RNA-seq Data Imputation using Generative Adversarial Networks", doi: 10.1101/837302.

[39] S. Yoon and S. Sull, "Gamin: Generative adversarial multiple imputation network for highly missing data," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2020, pp. 8453–8461. doi: 10.1109/CVPR42600.2020.00848.

[40] Y. Wang, D. Li, X. Li, and M. Yang, "PC-GAIN: Pseudo-label Conditional Generative Adversarial Imputation Networks for Incomplete Data," Nov. 2020, [Online]. Available: http://arxiv.org/abs/2011.07770

[41] R. Shahbazian and I. Trubitsyna, "DEGAIN as tool for Missing Data Imputation," 2023. [Online]. Available: http://ceur-ws.org

[42] J. Hwang and D. Suh, "CC-GAIN: Clustering and classification-based generative adversarial imputation network for missing electricity consumption data imputation," Expert Syst Appl, vol. 255, Dec. 2024, doi: 10.1016/j.eswa.2024.124507.

[43] W. Sudrajat, I. Cholid, and J. Petrus, "Wahyu Sudrajat et al, Penerapan Algoritma K-Means Untuk

[44] A. Khoirunnisa and N. G. Ramadhan, "Improving malaria prediction with ensemble learning and robust scaler: An integrated approach for enhanced accuracy," JURNAL INFOTEL, vol. 15, no. 4, pp. 326–334, Nov. 2023, doi: 10.20895/infotel.v15i4.1056.

[45] M. F. Dzulkalnine and R. Sallehuddin, "Missing data imputation with fuzzy feature selection for diabetes dataset," SN Appl Sci, vol. 1, no. 4, Apr. 2019, doi: 10.1007/s42452-019-0383-x.

[46] I. Gad, D. Hosahalli, B. R. Manjunatha, and O. A. Ghoneim, "A robust deep learning model for missing value imputation in big NCDC dataset," Iran Journal of Computer Science, vol. 4, no. 2, pp. 67–84, Jun. 2021, doi: 10.1007/s42044-020-00065-z.

[47] J. H. Li et al., "Comparison of the effects of imputation methods for missing data in predictive modelling of cohort study datasets," BMC Med Res Methodol, vol. 24, no. 1, Dec. 2024, doi: 10.1186/s12874-024-02173-x.