



ARTICLE

A Modified Deep Residual-Convolutional Neural Network for Accurate Imputation of Missing Data

Firdaus Firdaus, Siti Nurmaini*, Anggun Islami, Annisa Darmawahyuni, Ade Iriani Sapitri, Muhammad Naufal Rachmatullah, Bambang Tutuko, Akhiar Wista Arum, Muhammad Irfan Karim, Yultrien Yultrien and Ramadhana Noor Salassa Wandya

Intelligent System Research Group, Faculty of Computer Science, Universitas Sriwijaya, Palembang, 30139, Indonesia

*Corresponding Author: Siti Nurmaini. Email: siti_nurmaini@unsri.ac.id

Received: 10 July 2024 Accepted: 21 October 2024 Published: 17 February 2025

ABSTRACT

Handling missing data accurately is critical in clinical research, where data quality directly impacts decision-making and patient outcomes. While deep learning (DL) techniques for data imputation have gained attention, challenges remain, especially when dealing with diverse data types. In this study, we introduce a novel data imputation method based on a modified convolutional neural network, specifically, a Deep Residual-Convolutional Neural Network (DRes-CNN) architecture designed to handle missing values across various datasets. Our approach demonstrates substantial improvements over existing imputation techniques by leveraging residual connections and optimized convolutional layers to capture complex data patterns. We evaluated the model on publicly available datasets, including Medical Information Mart for Intensive Care (MIMIC-III and MIMIC-IV), which contain critical care patient data, and the Beijing Multi-Site Air Quality dataset, which measures environmental air quality. The proposed DRes-CNN method achieved a root mean square error (RMSE) of 0.00006, highlighting its high accuracy and robustness. We also compared with Low Light-Convolutional Neural Network (LL-CNN) and U-Net methods, which had RMSE values of 0.00075 and 0.00073, respectively. This represented an improvement of approximately 92% over LL-CNN and 91% over U-Net. The results showed that this DRes-CNN-based imputation method outperforms current state-of-the-art models. These results established DRes-CNN as a reliable solution for addressing missing data.

KEYWORDS

Data imputation; missing data; deep learning; deep residual convolutional neural network

1 Introduction

An electronic health record (EHR) is a digital file containing medical details, such as lab results, of an individual, and it's stored online [1,2]. Primarily, EHRs serve the purpose of establishing goals, devising patient care plans, recording care delivery, and evaluating the resulting outcomes [2]. Thus, it can be shared among various facilities and accessed swiftly by either patients or medical personnel. These records offer opportunities to improve patient care, integrate performance metrics into clinical practice, and streamline clinical research [2,3]. For instance, one research utilizing EHRs involves using



machine learning regression techniques to forecast cardiovascular risk [4]. Such predictive models can function as decision-support tools aiding doctors and healthcare providers in patient management and proactive interventions [5]. However, EHR datasets often exhibit a certain degree of incompleteness, with missing data affecting the quality of results and potentially introducing bias [6]. Missing data occurs when values for variables of interest are not recorded or measured for all subjects in the sample. There are several reasons for missing data, including patient refusal to complete certain questions, data acquisition equipment errors, and measurements with different sampling periods [7].

The most commonly used methods for estimating missing data generally fall into three categories [3]: deletion methods, methods for linear imputation based on statistics, and methods for non-linear imputation based on machine learning. Deletion approaches entail the direct removal of missing data, which, as prior research has demonstrated, can cause considerable changes in the analysis results due to the absence of critical information [8]. In contrast, statistically-based linear imputation methods involve replacing missing values with statistical measures such as the mean [9], or the most common value within a global or local range of the time series data [10]. While they use correlations between measurements across multiple data streams simultaneously, they ignore correlations within individual data streams [11]. As a result, these approaches may overlook important information. Furthermore, the advantages of these methods are the ease of implementation and interpretation, and the availability of various tools and techniques to support their use [12]. However, they often do not consider complex structures in the data and may produce biased or inaccurate results depending on the case at hand [13]. Alternatively, machine learning-based non-linear imputation methods can handle more complex correlations in the data, but they require greater computational resources and more difficult interpretation [14]. In practical use, method selection should consider the balance between practicality, accuracy, and interpretation of results. In the field of imputation missing data analysis, different strategies, such as deep learning (DL) algorithms have become more popular recently [15,16].

Alternative approaches such as deep learning (DL) methods are increasingly attracting attention in missing data imputation analysis [16]. This is because DL methods have proven to be able to take advantage of and manage large data sets with remarkable capabilities [17]. The Convolutional Neural Network (CNN) excels in capturing complex patterns and relationships within structured data [18]. In [19–21], several works have demonstrated the ability of CNNs to effectively process structured data, such as high-dimensional data, through the application of convolutional layers that focus on local features and hierarchical pattern recognition. However, CNN may struggle with imputing missing values if the data lacks spatial relationships, as the learning of kernel weights relies heavily on identifying spatial features. On the other hand, U-Net is specifically designed for spatial relationships that are essential for effective imputation [22]. However, if there isn't any discernible spatial relationship in the data, the kernel's learning process can go off course and provide inefficient imputation. The U-Net architecture was found to be less effective in managing missing data when the complex data structure did not align with the patterns that the U-Net is designed to recognize [22].

Current imputation methods, including traditional statistical methods and machine learning approaches like CNN and U-Net, have limitations in handling complex data structures and correlations within individual data streams. These methods either oversimplify the imputation process or require substantial computational resources without guaranteeing accurate results [18]. CNN and U-Net struggle to handle complex data structures and correlations within individual data streams effectively [18,22]. This gap underscores the need for a more advanced imputation method that can address these challenges by maintaining computational efficiency while improving accuracy, particularly in the context of medical datasets.

To address these limitations, we propose a novel neural network architecture called Deep Residual-Convolutional Neural Network (DRes-CNN). The DRes-CNN model modifies the foundation of the original Residual Neural Network (ResNet) architecture [22]. DRes-CNN integrates advanced techniques to enhance both feature extraction and learning capabilities. One key aspect of DRes-CNN is the incorporation of residual connections not only within individual layers but also across different network modules. This facilitates the flow of gradients during training, alleviating the vanishing gradient problem and enabling more efficient optimization. Additionally, by promoting information flow across the network, these connections enable better preservation of useful information throughout the deep layers, ultimately improving the network's ability to learn intricate patterns [23]. We also explore two different neural network architectures, including Low Light-Convolutional Neural Network (LL-CNN) and U-Net to illustrate how the imputation process's use of time information inclusion has improved.

The purpose of this study is to evaluate the robustness of our imputation method and fill in the missing information. The main conclusions of our research demonstrate how well DRes-CNN imputes missing values in medical information datasets with remarkable accuracy. The research presents novel findings in four areas:

- **Development of a novel DRes-CNN architecture:** We propose an advanced Deep Residual-Convolutional Neural Network (DRes-CNN) specifically designed to improve data imputation accuracy, addressing the limitations of existing methods.
- **Comprehensive evaluation of diverse datasets:** Our approach is tested on three publicly available datasets (MIMIC-III, MIMIC-IV, and the Beijing Multi-Site Air Quality dataset) with various types of missingness, ensuring robustness across different scenarios.
- **Application to real-world healthcare data:** The proposed DRes-CNN model is implemented for multivariate data imputation in medical applications, demonstrating its practical relevance in clinical settings.
- **Superior performance in imputation and prediction tasks:** Quantitative comparisons show significant improvements in RMSE over existing state-of-the-art methods, confirming the effectiveness of our approach in both imputation and downstream prediction tasks

2 Related Works

Missing data imputation has been a longstanding challenge in various domains. Traditional approaches often rely on statistical methods or simple imputation techniques, which can introduce bias if assumptions about the data distribution are incorrect. In recent years, deep learning-based methods have emerged as promising alternatives, offering more flexible and powerful approaches.

Previous studies have explored a range of techniques for handling missing data. For instance, autoencoder models [24] aim to learn representations that can reconstruct the original data, minimizing reconstruction errors. This study used clinical data to train the power of the model. However, for large and complex datasets this model cannot yield promising effectiveness. Similarity-aware diffusion Model-Based Imputation (SADI) [25] leverages diffusion models and self-attention to effectively impute missing values in EHRs. Current models typically rely on correlations between time points and features, which is effective for data with strong correlations, such as Intensive Care Unit (ICU) data. However, this method is less effective for non-ICU data with weak correlations. This study uses the Critical Path for Alzheimer's Disease (CPAD) and PhysioNet Challenge 2012 datasets. The Similarity-Aware Diffusion Model-Based Imputation (SADI) method demonstrates superior performance compared to current state-of-the-art models. SADI, which leverages diffusion

models and a self-attention mechanism, is effective in imputing missing data under various missingness mechanisms, including Missing Completely At Random (MCAR), Missing At Random (MAR), and Missing Not At Random (MNAR), and is better at improving accuracy and reducing bias. Applying the clustering process to big datasets with many of patients can be computationally expensive, which is one of SADI's limitations. In [26,27], using the method of multiple imputation by chained equations (MICE), a chained approach is used to mine all relevant evidence of missing values and to further estimate those missing values. The datasets used in this study include the Longitudinal Anthropometric Data [26] and the PFA-Datasets in Clinic from [27]. However, in the case of very large datasets, machine learning-based imputation approaches require significant computation time, making these approaches less practical for the task of imputing a large amount of missing data.

In Generative Adversarial Networks (GANs) [28,29] have also been applied to missing data imputation, generating realistic samples to fill in missing values. GANs can suffer from mode collapse, where the generator produces limited diversity in the imputed data. In clinical settings, where variability and accuracy are crucial, mode collapse can lead to biased or incomplete imputations that do not adequately represent the range of possible values. These weaknesses make GANs less suitable for clinical data imputation compared to other methods that might offer more stability, transparency, and accuracy. CNNs have also been studied in [30] for the imputation of missing data, utilizing their ability to find spatial correlations and nonlinearities within the data. CNNs have the ability to use techniques such as fuzzy c-means clustering to group data into clusters according to membership values. As a result, it is possible to arrange the data in a way that captures robust attribute relationships. Multiple modalities' worth of data can be handled and imputed simultaneously by CNNs. However, because the training process is complex and precise hyperparameter tweaking is necessary, imputation tasks may be challenging to comprehend. Due to its complexity, the model may be less useful for real-world applications, particularly in situations where computing resources are scarce or when access to professionals with the necessary experience to fine-tune such models is difficult.

In contrast to existing methods, our proposed approach, DRes-CNN, leverages deep residual-based CNNs to extract deep features from structured data and impute missing values using a learned kernel. This novel approach offers promising results, outperforming existing methods in key performance metrics. This novel approach offers promising results, outperforming existing methods in key performance metrics. Table 1 for details of the conclusions of several approaches to methods used for handling missing data.

Table 1: Summary of related works

Method	Problem handled	Dataset used	Methodology	Results	Limitations
Autoencoder	Missing value imputation	Incomplete Datasets in Clinical Information	Reconstruction-based	Minimizes reconstruction errors	Can be complex for large datasets

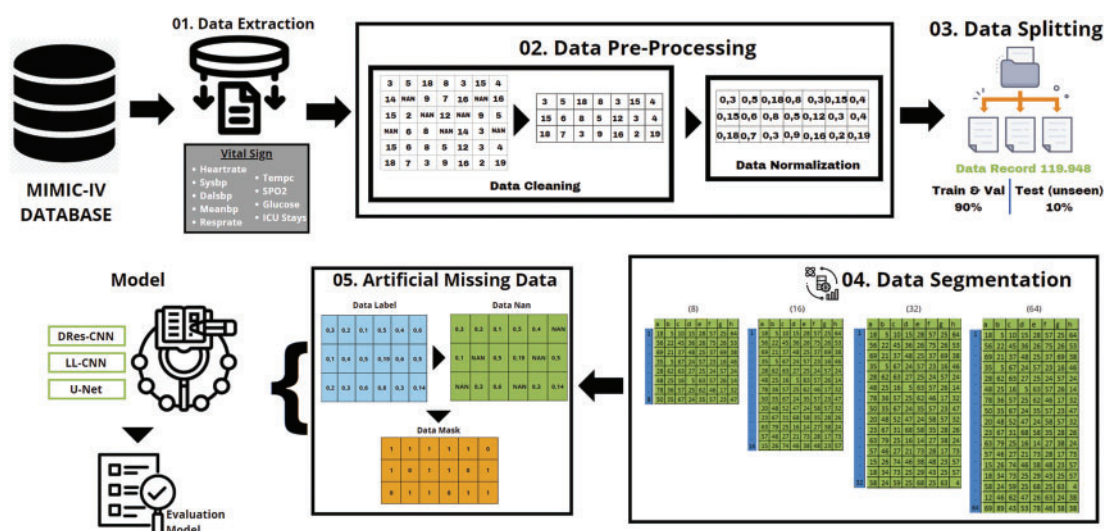
(Continued)

Table 1 (continued)

Method	Problem handled	Dataset used	Methodology	Results	Limitations
SADI	Missing data imputation in EHR	CPAD, PhysioNet Challenge 2012	Diffusion models, self-attention	Superior performance, effective under various missingness mechanisms	Can be computationally expensive for large datasets
MICE	Missing data imputation	Longitudinal Anthropometric Data, PFA-Datasets	Chained equations	Estimates missing values	Can be computationally expensive for large datasets
GANs	Missing data imputation	Traffic Dataset, EHR dataset	Generative adversarial networks	Learns to reconstruct original input values	Can be computationally expensive due to training requirements
CNNs	Missing data imputation	EHR Dataset	Spatial correlations, nonlinearities	Effective for handling multiple modalities	Complex training process, hyperparameter tuning

3 Material and Method

We provide a brief overview of the datasets utilized in the study in this section, the benchmarking exercise, and the method for assessing the quality of imputation illustrated in Fig. 1.

**Figure 1:** The research methodology

3.1 Data Preparation

Vital signs and pertinent data taken from the MIMIC-IV database were used in our investigation [31,32]. We received permission to extract data from the MIMIC-IV for research purposes (Certification Number: 50640755) and completed the web-based training course offered by the National Institutes of Health to ensure the safety of human study participants before gaining access to the database. High-quality data from 2008 to 2019 is included in the publicly accessible MIMIC-IV database, which was created by the review boards of Beth Israel Deaconess Medical Center and Massachusetts Institute of Technology (MIT), both located in Cambridge, Massachusetts. MIMIC-IV contains extensive patient data, including demographics, hospitalization records, diagnoses made under the International Classification of Diseases, Ninth Revision (ICD-9), laboratory tests, prescription drugs, procedures, fluid balance, discharge summaries, bedside vital sign measurements, treatment notes, radiology reports, and survival data. The MIMIC-IV database consists of numerous linked tables that use standardized ICD-9 numbers to describe the data of each ICU patient.

3.2 Data Preprocessing

Data preprocessing in this study entails some procedures and techniques meant to handle missing or insufficient data, as depicted in Fig. 1. The primary procedures for Preprocessing data for imputation usually entail: Missing Data Detecting: Identifying that attributes or variables within the dataset contain missing values. To approximate missing values, one can do the following: remove rows or columns containing missing values; replace missing values with the mean of the available values within the corresponding attribute; create predictive models to estimate missing values based on relationships with other variables, and utilize statistical distributions to create random values that roughly match the distribution of the available data. Use a large query based on a specific value from the chart events table of the MIMIC-IV database in the itemid column to obtain the vital signs dataset of ICU patients.

- The total raw data from the ICU patient vital signs dataset is 7.786.008 with missing data (Table 2), has a unique id or stay_id for each patient from 38675535 to 37145532, and the duration of each patient's stay in the ICU from 2182-01-25 07:00:00+00:00 to 2146-04-27 09:33:00+00:00.

Table 2: Total information from the MIMIC-IV vital sign dataset for each column before removing missing data

Column	Number of data	Missing data	Min	Max
Heart rate	6,458,181	1,327,827	1	295
sbp	2,263,830	5,522,178	1	357
dbp	2,263,308	5,522,700	1	299
mbp	2,434,351	5,351,657	1	299
Resprate	6,455,503	1,330,505	0.13	69
Temperature	1,776,877	6,009,131	15	43.11
spo2	6,323,768	1,462,240	0.01	100
Glucose	1,325,699	6,460,309	0.12	1,276,100

Our proposed model analyzes eight vital sign features: heart rate, systolic blood pressure, diastolic blood pressure, mean blood pressure, respiratory rate, body temperature, oxygen saturation, and glucose level. We used an inner join to get a more extensive dataset. Consequently, our final dataset contained 119,948 unique ICU patients (vital signs) as a result (Table 3). This entire dataset includes the min and max data of vital signs recorded during a patient's ICU stay.

Table 3: The number of data for the column from vital sign MIMIC-IV dataset after removing missing data

Column	Number of data	Min	Max
Heart_rate	119,948	23	204
sbp	119,948	19	304
dbp	119,948	3	234
mbp	119,948	6	238
Resp_rate	119,948	1	69
Temperature	119,948	29.3	41.5
spo2	119,948	10	100
Glucose	119,948	1	500036

- Standardization, after imputation of the missing data the dataset was normalized using the Min-Max Scaler [33,34], this procedure is vital to guarantee uniformity or standardization among the features. The following Eq. (1) illustrates the data normalization:

$$x_{norm} = \frac{x_{inp} - x_{min}}{x_{max} - x_{min}} \quad (1)$$

where x_{norm} denotes the normalized data; x_{inp} denotes the input value; x_{min} denotes the minimum value of the feature; and x_{max} denotes the maximum value of the feature.

- In this step, concerning MIMIC-IV datasets, a data adjustment process ensues, aiming to extract the minimum volume of data from each icustay_id. The meticulous step serves as the foundation for generating multiple data segments from every icustay_id, which is pivotal for subsequent analysis or modeling endeavors. The segmentation strategy employed entails categorizing the data into various subsets, comprising 8, 16, 32, and 64 groups based on the Eq. (2) as follows:

$$\text{Number of segments} = \text{data length per stay_id} - \text{segment group} + 1 \quad (2)$$

3.3 Create Artificial Missing Data

We have a dataset structured as $m \times n$ matrix, where n stands for features and m for cases. The training data ($X_{raw}^t \in \mathbb{R}^{m \times n}$) and the testing data ($X_{raw}^{t'} \in \mathbb{R}^{m' \times n}$) are two halves of the original data set. We intentionally manufactured the missing data in ($X_{raw}^t \in \mathbb{R}^{m \times n}$) and ($X_{miss}^{t'} \in \mathbb{R}^{m' \times n}$), where duplicate data was initially provided with these missing values (Fig. 2). This was done following the division of data to produce training and test sets. New versions of the training and testing data ($X_{miss}^t \in \mathbb{R}^{m \times n}$) and ($X_{miss}^{t'} \in \mathbb{R}^{m' \times n}$), respectively, are obtained when the missingness value is created. To add missing values to the clean test data during the testing phase, ($X_{miss}^{t'} \in \mathbb{R}^{m' \times n}$), we repeat the same training process, resulting in the corrupted test data ($X_{miss}^{t'} \in \mathbb{R}^{m' \times n}$). Subsequently, after training the DRes-CNN (refer to Algorithm 1), we assess its performance on ($X_{miss}^{t'} \in \mathbb{R}^{m' \times n}$) to impute the missing data, as depicted in Fig. 1. Similarly, after the completion of the imputation process in our scenario, artificial data emerges

wherein all values have been altered due to the convolution operation. Therefore, as shown in Fig. 1, we use masking on the newly created data matrix to recover the imputed values.

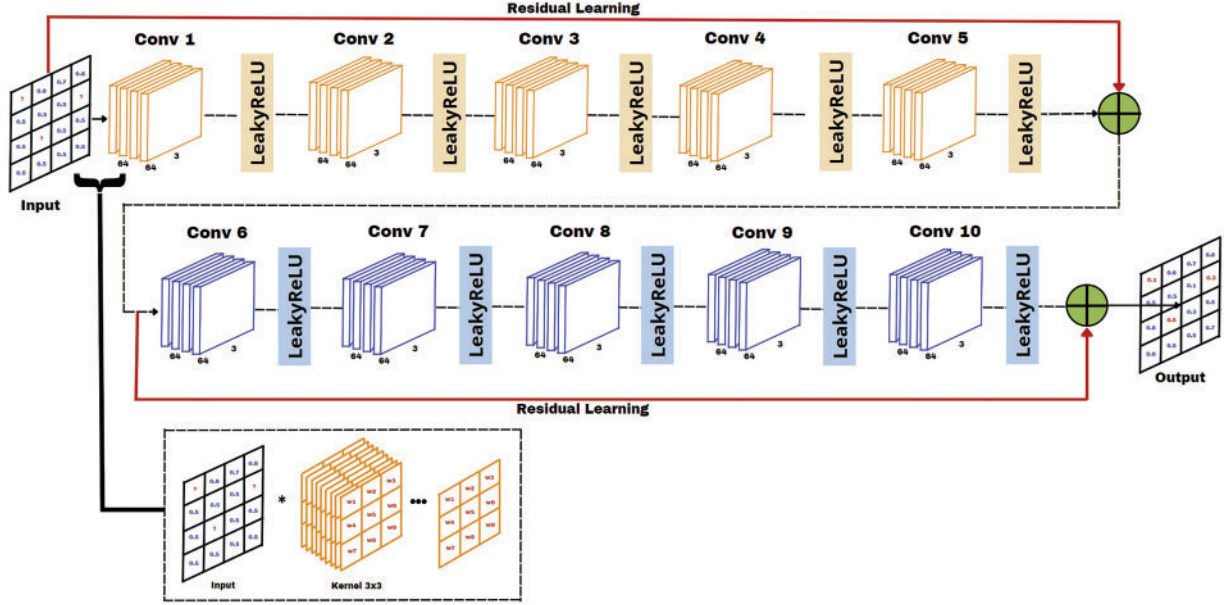


Figure 2: Proposed DRes-CNN architecture

Algorithm 1: The approach for data imputation

INPUT:

$X_{raw}^t \in$: Raw data matrix with missing data, where mmm is the number of rows and n is the number of columns.

OUTPUT:

$Y_{Pred}^t \in$: Imputed data prediction

Begin

Step 1:

1. Clean Data:

- $X_{cleaned} \leftarrow \text{DROP_MISSING_DATA}(X_{raw}^t)$
- Clean data by dropping rows containing missing data

2. Calculate Column Means:

- $X_{mean} \leftarrow \text{CALCULATE_COLUMN_MEAN}(X_{cleaned})$
- Calculated the mean for each column in the cleaned data

3. Normalize Data:

- $\text{NORMALIZE}(X_{imputed})$
- Standardize each column in $X_{imputed}$ using MinMaxScaler // normalization

(Continued)

Algorithm 1 (continued)**4. Split Data:**

- $X_Splitted \leftarrow \text{SPLIT_DATA}(X_{raw}^t)$
 - Split the raw data into the training data ($X_{raw}^t \in \mathbb{R}^{m \times n}$) and the testing data ($X_{raw}^{t'} \in \mathbb{R}^{m' \times n}$)

5. Impute Missing data

- $X_imputed \leftarrow \text{FILL_MISSING_DATA}(X_{miss}^t, X_mean)$
 - Complete the blank values in the row data with the previously computed means

Step 2:**1. for every column of x in $X_imputed$**

$x \leftarrow \text{COMPUTE_mean}(x)$
 compute the mean for each column
 $\text{REPLACE_MISSING_DATA}(?, x)$
 substitute missing data with the computed mean

2. Duplicate Columns:

- $X_{miss}^{t'} \leftarrow \text{DUPLICATE_COLUMNS}(X_imputed)$
 - Replicate columns in $X_imputed$ to obtain ($X_{miss}^{t'} \in \mathbb{R}^{m' \times n}$)

3. Predict Values:

- $Y_{Pred}^t \leftarrow \text{PREDICT}(\text{DRes-CNN}, X_{miss}^{t'})$
 - Predict the value using the trained DRes-CNN Model

end for**End****3.4 The Proposed Model DRes-CNN**

The proposed method utilizes a specific type of neural network architecture known as Convolutional Neural Networks (CNN). Our network architecture, which is a dense convolutional neural network with residual connections (DRes-CNN), is depicted in Fig. 2. This DRes-CNN network structure resembles the original CNN architecture for image super-resolution [20], but is modified to fit tabular or time-series data in capturing complex patterns and relationships within structured data [18]. This architecture is made to maximize the information flow between network levels using a straightforward connecting scheme. All other levels are directly connected. Every layer passes its feature map to all the following levels and receives extra input from all prior layers to preserve the feed-forward character of the system. Dense networks are one type of CNN that connects each layer in a feed-forward way. These dense networks increase feature propagation, stimulate feature reuse, and solve the vanishing-gradient problem in contrast to typical convolutional networks, where each layer is merely connected to the next layer.

The convolutional layers in our proposed DRes-CNN architecture are specifically designed to detect dependencies between features in the EHR dataset, both temporally and contextually. By employing dense and residual connections, the model enhances feature propagation, reuses information from earlier layers, and improves the learning process for long-range feature dependencies. Dense and residual connections function differently. While feature maps in residual connections are combined by summation before being transmitted to a specific layer, feature maps in dense connections are merged based on all prior feature maps. The benefit of having both kinds of connections is that, in a dense connection, all of the features from the prior convolution block may be used, whereas, in a residual connection, we are limited to using one historical feature map.

As can be seen in Fig. 2, the suggested DRes-CNN architecture consists of two residual connections and ten convolutional layers with progressively more filters. The residual connection in Fig. 2 is the solid line that carries the input layer to the summing operator. In ideal circumstances, a deep neural network with more layers can outperform shallower networks and drastically minimize error rates. To outperform a straightforward deep neural network, we incorporate residual blocks.

The architecture in our model has 64 filters with multiple kernel sizes of 3×3 , followed by the LeakyReLU activation function. After multiple convolution layers, there is residual linking which is done by adding the output of the first layer to the output of the last layer before the next convolution layer. This architectural design helps in learning more detailed features and retains the original information of the input. This study, provided detailed descriptions of our architecture, specifying that it consists of ten convolutional layers with both dense and 2 residual connections. We explained the advantages of using both connection types, as dense connections facilitate the utilization of features from previous convolutional blocks, while residual connections enhance the flow of information by summing the outputs of earlier layers with later ones. We also justified the choice of architecture by noting that deeper networks tend to outperform shallower counterparts, especially when incorporating residual blocks that minimize error rates. The total parameters in the model amount to 8.545, which also includes the weights and biases in each convolution layer. To offer transparency, we included Table 3, which outlines the architecture modification information, detailing output shapes and the number of parameters for each layer. The architecture in our model has modification information shown in Table 4.

Table 4: Architecture modification information

Convolution	Layer (type)	Output shape	Parameters number	Connected to
Input data	Input_1 (InputLayer)	[(None, 64, 64, 1)]	0	[]
Conv 1	conv2d (Conv2D)	[(None, 64, 64, 64)]	128	['input_1 [0][0]']
Head	leaky_re_lu (LeakyReLU)	[(None, 64, 64, 64)]	0	['conv2d [0][0]']
Conv 2	conv2d_1 (Conv2D)	[(None, 64, 64, 64)]	4160	['leaky_re_lu [0][0]']
	leaky_re_lu_1 (LeakyReLU)	[(None, 64, 64, 64)]	0	['conv2d_1 [0][0]']
Conv 3	conv2d_2 (Conv2D)	[(None, 64, 64, 64)]	4160	['leaky_re_lu_1 [0][0]']
	leaky_re_lu_2 (LeakyReLU)	[(None, 64, 64, 64)]	0	['conv2d_2 [0][0]']
Conv 4	conv2d_3 (Conv2D)	[(None, 64, 64, 64)]	4160	['leaky_re_lu_2 [0][0]']
	leaky_re_lu_3 (LeakyReLU)	[(None, 64, 64, 64)]	0	['conv2d_3 [0][0]']

(Continued)

Table 4 (continued)

Convolution	Layer (type)	Output shape	Parameters number	Connected to
Conv 5	conv2d_4 (Conv2D)	[(None, 64, 64, 64)]	4160	['leaky_re_lu_3 [0][0]']
	leaky_re_lu_4 (LeakyReLU)	[(None, 64, 64, 64)]	0	['conv2d_4 [0][0]']
Residual	add (Add)	[(None, 64, 64, 64)]	0	['leaky_re_lu [0,0]', 'leaky_re_lu_4 [0][0]'] ['add [0][0]']
Conv 6	conv2d_5 (Conv2D)	[(None, 64, 64, 64)]	4160	
	leaky_re_lu_5 (LeakyReLU)	[(None, 64, 64, 64)]	0	['conv2d_4 [0][0]']
Conv 7	conv2d_6 (Conv2D)	[(None, 64, 64, 64)]	4160	['leaky_re_lu_4 [0][0]']
	leaky_re_lu_6 (LeakyReLU)	[(None, 64, 64, 64)]	0	['conv2d_5 [0][0]']
Conv 8	conv2d_7 (Conv2D)	[(None, 64, 64, 64)]	4160	['leaky_re_lu_5 [0][0]']
	leaky_re_lu_7 (LeakyReLU)	[(None, 64, 64, 64)]	0	['conv2d_6 [0][0]']
Conv 9	conv2d_8 (Conv2D)	[(None, 64, 64, 64)]	4160	['leaky_re_lu_6 [0][0]']
	leaky_re_lu_8 (LeakyReLU)	[(None, 64, 64, 64)]	0	['conv2d_7 [0][0]']
Conv 10	conv2d_9 (Conv2D)	[(None, 64, 64, 64)]	4160	['leaky_re_lu_8 [0][0]']
Residual	add_1(Add)	[(None, 64, 64, 64)]	0	['leaky_re_lu_6 [0][0]', 'leaky_re_lu_10 [0][0]']

Residual connections are introduced after Conv 5 and Conv 10, which are created by adding a convolutional layer's output to a preceding layer's output before the layer's activation function is sent through (Table 3). A situation with randomly generated missing data in the input data matrix and the same whole data matrix produced at the output is shown in Fig. 3. Assume that we have a kernel with dimensions $M \times N$ (where M represents the number of rows and N denotes the number of columns), and there's a missing value for a unit O_{ij} within the feature map. An input-weighted total within a given patch size $M \times N$ is computed to find the value of O_{ij} . These kernel weights are adjustable parameters that are learned by utilizing the linear characteristics present in the given data. Once optimized during the training phase, these weights can then be employed for imputing missing data during the testing phase. Eq. (3) illustrates the process by which a filter convolves over the data and imputes the missing data O_{ij} .

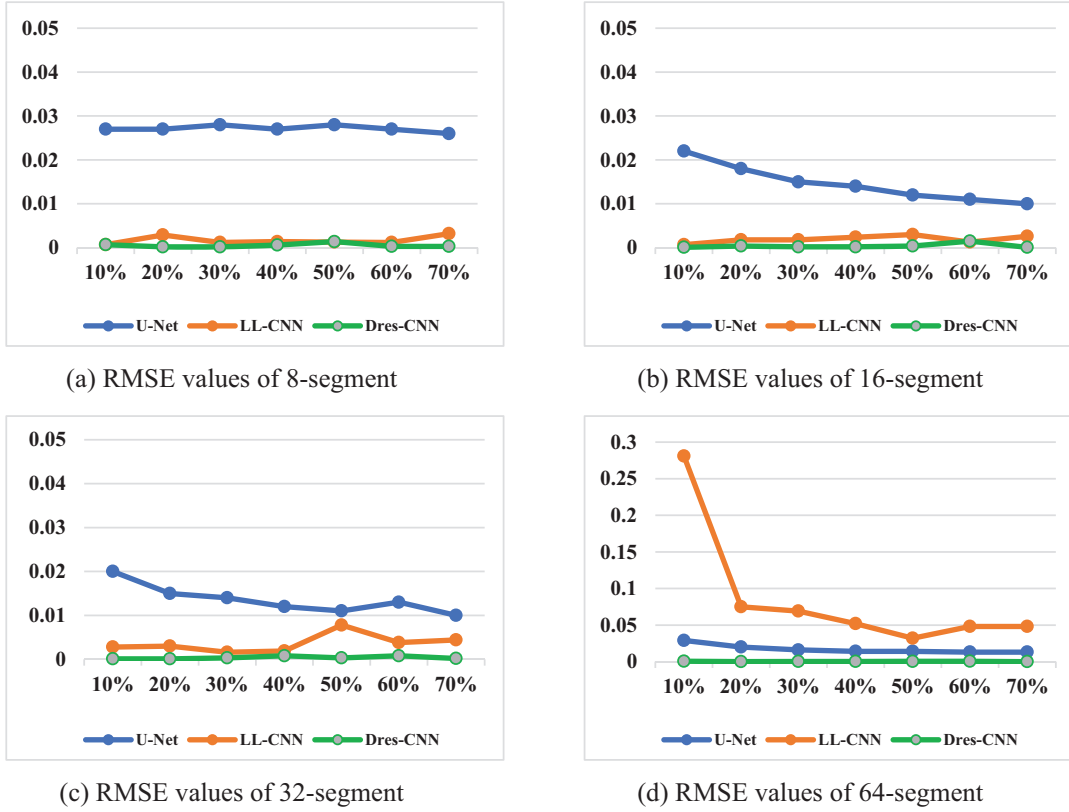


Figure 3: The performance analysis of RMSE (MIMIC-IV) with U-Net, LL-CNN and DRes-CNN. The x-axis shows the missing ratio datasets and the y-axis shows the RMSE

$$O_{ij} = f \left(\sum_m \sum_n I_{i+m,j+n} K_{m,n} + b \right) \quad (3)$$

where $K_{m,n}$ is the weight (kernel) at position (m,n) used for convolution. b representing the bias term. f represents a residual function. This process involves the summation of the dot product of each position in the input data between the input and the kernel weights, which is then added with a bias. The mathematical function for LeakyReLU as shown in Eq. (4).

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0 \end{cases} \quad (4)$$

where α is a small positive value. Thus, when x is less than 0, the output is αx , while when x is greater than or equal to 0, the output is x . LeakyReLU is chosen as the activation function due to its effectiveness and widespread use in practical applications. Additionally, LeakyReLU can help neural networks learn patterns and relationships between features in the dataset. Usually in ReLU, negative input values become zero, while positive values do not change. However, in LeakyReLU, when the input is negative, it does not become zero but is scaled by a small positive value Eq. (4). This can help overcome the “dead neuron” problem that can occur in ReLU, where the neuron saturates at negative values and does not update its weights during training. After several layers of convolution, there is a residual linking which is done by adding the output of the first layer to the output of the last layer

before the next convolution layer. The imputation of a randomly missing value in a data sample that goes through a trained model is further demonstrated in Fig. 3. Once the model undergoes training, it gains the capability to impute missing values through the learned kernels.

The evaluation process entails using the available data to train the network to predict missing data, with the model being adaptable for tabular or time series data. Important features are preserved across levels by using jump connections. Several hyperparameters, such as the Adam optimizer, 200 epochs, batch sizes of 16 and 32, and a learning rate of 0.001, are used to maximize the model's performance. The model's activation function is linear, and the loss function is a mean squared error. By partitioning the dataset into multiple subsets, we were able to validate the model's performance across different data segments, thereby enhancing the generalizability of the results. Additionally, a learning rate schedule based on exponential decay is employed to enhance the training. Algorithm 2 presents the pseudocode for the DRes-CNN algorithm.

Algorithm 2: Dres-CNN data imputation process

INITIALIZE NETWORK ARCHITECTURE

INPUT:

- $X_{raw}^t \in$: Data matrix with missing data
- 10 Convolutional layers with increasing filters and kernel size 3×3
- LeakyReLU activation function after each convolution layer
- Residual connections after Conv5 and Conv10

BEGIN

Define Convolutional Block

```
conv_block(input, filters, kernel_size):
    output = conv(filters = filters, kernel_size = kernel_size, padding='same')(input)
    output = LeakyReLU( )
    return  $X_{raw}^t$ 
```

Define residual block

```
residual_block(input1, input2):
    return Add()([input1, input2])
```

Define DRes-CNN model

```
DRes_CNNs(input_shape):
    inputs = Input(shape = input_shape)
    x = conv_block(inputs, 64, (3, 3))
    conv1_output = x
    for i in range(4):
        x = conv_block(x, 64, (3, 3))
    conv5_output = x
    residual1_output = residual_block(conv1_output, conv5_output)
    for i in range(4): // Layers 6 to 9
        x = conv_block(residual1_output, 64, (3, 3))
    conv10_output = x
    residual2_output = residual_block(residual1_output, conv10_output)
    final_output = Conv2D(filters = 1, (3, 3), activation = 'sigmoid', padding =
'same')(residual2_output)
    model = model (inputs, final_output)
    return model
```

(Continued)

Algorithm 2 (continued)**Training Procedure**

```

for epoch in range(num_epochs):
    for batch in generate_batches(train_data, batch_size):
        inputs, targets = batch
        with tf.GradientTape() as tape:
            predictions = model(inputs)
            loss = mean_squared_error(targets, predictions)
            grads = tape.gradient(loss, model.trainable_weights)
            optimizer.apply_gradients(zip(grads, model.trainable_weights))
    evaluate_model(model, val_data)

```

END

During training, the learning rate of the model can be adjusted to facilitate a more efficient convergence to the optimal solution. Combining optimization methods with hyperparameters improves the performance of the DRes-CNN algorithm. The Intel® Core™ i9-9920X CPU running at 3.50 GHz, 490,191 MB of RAM, and GeForce 2080 RTX Ti from NVIDIA Corporation GV102 (rev a1) were the characteristics of the personal computer (PC) used to train the network. Pytorch 1.7.1 package together with Python were used to create the network.

3.5 Model Evaluation

Model evaluation in filling data with DRes-CNN involves evaluating how the trained deep learning model is used to predict missing values in the dataset. Model performance evaluation uses the root mean squared error (RMSE) metric to assess the average error produced by the model [9]. The smaller the value of mean absolute error (MAE) and RMSE, the better the performance of our imputation method. MAE and RMSE are used as model evaluation metrics [9] and are described in Eqs. (5) and (6).

$$MAE = \frac{1}{n} \sum_{j=1}^n |(y_{act} - \widehat{y_{pred}})| \quad (5)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_{act} - \widehat{y_{pred}})^2} \quad (6)$$

where y_{act} is the actual value, $\widehat{y_{pred}}$ is the predicted values by the regression model, and n number of observations/rows [9].

4 Experimental Results and Discussions**4.1 Experimental Research**

This section presents the study results of our suggested approach and evaluates the significant improvement in handling missing data performance compared to other methods. We use the MIMIC-IV dataset described in Section 3.2. At first, data cleaning for missing data are randomly added, replacing the missing values with zero. Next, our proposed DRes-CNN approach is used to impute those missing values, and other imputation techniques, such as U-Net and LL-CNN, are selected for comparison purposes. Moreover, the robustness of the DRes-CNN approach is tested with various missing ratios (10%, 20%, 30%, 40%, 50%, 60%, and 70%) on the MIMIC-IV dataset. We also evaluated the imputation performance of DRes-CNN by using three supervised learning algorithms

to train classifiers on the imputed data. Experiments were conducted three times with random data partitioning (90% for training and 10% for testing), and the average classification accuracy along with the standard deviation was calculated as the final performance score. Initially, 10% of attribute values from all clean data were artificially removed to generate a dataset with missing values.

4.2 Result

This section provides a thorough examination of comparing the proposed technique with alternative imputation methods. Table 4 illustrates the assessment outcomes (RMSE) acquired through various approaches when dealing with randomly generated missing values, alongside other advanced imputation methods like U-Net, LL-CNN, and DRes-CNN. Experimental results were obtained through various missing ratios (i.e., 10%–70%). It can be seen in Table 5 that the a substantial difference in performance between the proposed method and the others. Notably, regardless of the missing ratio, the RMSE results remain relatively stable, indicating that the proposed approach maintains its efficacy even when the level of missing data increases. Fig. 3, a graphical depiction of Table 5, makes this distinction easy to see. The best value for each dataset is displayed in a column of Table 4 in bold type. Particularly, the superiority of the DRes-CNN method is underscored, showcasing its adeptness in accurately imputing missing values within vital signs data, even under high missing data scenarios. Our results demonstrated that the Dres-CNN model consistently outperformed the other models, particularly when handling higher ratios of missing data. This performance can be attributed to key architectural enhancements, such as the use of skip and residual connections, which allow the model to retain both high-level and low-level features across layers. This retention is critical for capturing complex spatial relationships and non-linear dependencies, which are essential for accurate imputation.

Table 5: RMSE from U-Net, LL-CNN, and DRes-CNN with varying the ratio of missing value

Method	Segment	Batch size	10%	20%	30%	40%	50%	60%	70%
U-Net	8	16	0.02711	0.02730	0.02982	0.02691	0.02810	0.02785	0.02794
		32	0.02745	0.02707	0.02794	0.02723	0.02707	0.02658	0.02666
	16	16	0.03526	0.01922	0.01527	0.01686	0.01504	0.01202	0.01302
		32	0.02266	0.01805	0.01551	0.01443	0.01286	0.01181	0.01088
	32	16	0.03245	0.01846	0.01665	0.01288	0.01203	0.01290	0.01263
		32	0.02048	0.01593	0.01441	0.01306	0.01148	0.01296	0.01048
	64	16	0.02943	0.02095	0.01761	0.01618	0.01433	0.01418	0.01409
		32	0.03453	0.02149	0.01667	0.01478	0.01408	0.01303	0.01346
	8	16	0.00073	0.00293	0.00119	0.00141	0.00763	0.00129	0.00643
		32	0.00225	0.00505	0.00328	0.00239	0.00134	0.00284	0.00323
LL-CNN	16	16	0.00236	0.00196	0.00605	0.00360	0.00524	0.00413	0.00268
		32	0.00075	0.00184	0.00183	0.00244	0.00302	0.00135	0.00307
	32	16	0.00469	0.00718	0.00169	0.00564	0.00928	0.00388	0.01052
		32	0.00281	0.00300	0.00186	0.00192	0.00782	0.01108	0.00449
	64	16	0.28100	0.07554	0.06900	0.05217	0.06351	0.06351	0.07555
		32	0.31709	0.08878	0.38263	0.17603	0.03277	0.04808	0.04842

(Continued)

Table 5 (continued)

Method	Segment	Batch size	10%	20%	30%	40%	50%	60%	70%
DRes-CNN	8	16	0.00075	0.00048	0.00124	0.00081	0.00189	0.00131	0.00031
		32	0.00098	0.00024	0.00020	0.00064	0.00144	0.00035	0.00062
	16	16	0.00011	0.00146	0.00022	0.00045	0.00065	0.00155	0.00167
		32	0.00035	0.00045	0.00062	0.00029	0.00047	0.00173	0.00008
	32	16	0.00017	0.00018	0.00148	0.00081	0.00436	0.00085	0.00153
		32	0.00039	0.00128	0.00030	0.00156	0.00037	0.00128	0.00028
	64	16	0.00057	0.00006	0.00018	0.00020	0.00039	0.00055	0.00007
		32	0.00065	0.00053	0.00022	0.00042	0.00122	0.00722	0.00098

The experimental findings showcased that the several models, trained used batch size of 16, 64 segments, a 20% missing ratio, and undergoing 200 epochs, attained an impressive on the MIMIC-IV dataset (as depicted in Table 6). Through scrutinizing the model's efficacy across varied segment sizes and missing rates, we discern its capability to address an array of data patterns and handle missing data occurrences that mirror real-world scenarios, wherein data may manifest varying levels of incompleteness.

Table 6: The best model from U-Net, LL-CNN, and DRes-CNN

Segment	U-Net	LL-CNN	DRes-CNN
8	0.02658	0.00073	0.00020
16	0.01088	0.00075	0.00008
32	0.01048	0.00169	0.00017
64	0.01303	0.03277	0.00006

From the data presented in Tables 5 and 6, it's evident that DRes-CNN surpasses both U-Net and LL-CNN in performance. The model's ability to generate low RMSE suggests that the imputed data closely aligns with the true values missing from the dataset. A primary factor contributing to DRes-CNN superiority in data imputation is its tailored design for data processing tasks, where missing or corrupted data is common. Also, we conduct a comparison of each attribute from U-Net, LL-CNN, and DRes-CNN (Table 7).

Table 7: RMSE results of each attribute from U-Net, LL-CNN, and DRes-CNN

Vital signs	U-Net		LL-CNN		DRes-CNN	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Heartrate	0.009	0.0078	0.058525	0.070074	0.0009866300	0.0009865
Sysbp	0.008	0.0093	0.053339	0.066378	0.0009866411	0.0009866
Diasbp	0.007	0.0095	0.017790	0.025735	0.0009866883	0.0009866

(Continued)

Table 7 (continued)

Vital signs	U-Net		LL-CNN		DRes-CNN	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Meanbp	0.006	0.0099	0.030178	0.045168	0.0009866659	0.0009866
Resprate	0.011	0.0101	0.047702	0.061603	0.0009866502	0.0009866
Temps	0.007	0.0091	0.021124	0.025127	0.0009864810	0.0009864
Spo2	0.010	0.0134	0.015266	0.015728	0.0009861991	0.0009861
Glucose	0.009	0.0146	0.045483	0.045484	0.0009868535	0.0009868

Table 6 presents a comparison of the RMSE values of three different models: U-Net, LL-CNN, and DRes-CNN, in predicting various vital signs attributes. The analysis results show that DRes-CNN consistently has a lower RMSE compared to U-Net and LL-CNN for each attribute. This indicates that DRes-CNN tends to perform better in predicting vital signs values. However, there were variations in model performance between attributes, where some attributes had smaller RMSE than others, such as “spo2” which had relatively low RMSE for all models. The lower the RMSE value, the closer the model prediction is to the true value, so models with lower RMSE tend to have more accurate predictions. A direct comparison between the models can provide insight into the relative merits of each in predicting certain attributes, although overall DRes-CNN dominates. The best model to use for the vital sign prediction challenge can be chosen with the help of this analysis, as well as highlighting areas where improvements or adjustments may be needed.

4.3 Discussions

The Ablation Study

In this study, we conducted an ablation study to evaluate the performance of our proposed DRes-CNN model for handling missing data in structured datasets. The study focused on assessing the impact of different hyperparameters, and missing data ratios. We tested the model with varying missing data percentages (10% to 70%) and systematically tuned hyperparameters such as segment size (8, 16, 32, 64), batch size (16, 32), and learning rate (0.001), as well as the optimizer (Adam) and the number of epochs (200), to determine their influence on imputation accuracy. A comprehensive experiment and a detailed comparison of the results are provided in the subsequent section with an in-depth analysis shown in Fig. 4.

The performance of each model was assessed using the parameter RMSE. Our results demonstrate that the DRes-CNN model outperforms traditional methods, such as LL-CNN and U-Net, especially when dealing with higher missing data ratios. First, we evaluated the effect of different missing ratios, ranging from 10% to 70%. The results showed that the model performed best at lower missing ratios (around 10%), with performance gradually decreasing as the missing ratio increased. This indicates that the model is more efficient at handling datasets with fewer missing entries. Next, we examined segment sizes (8, 16, 32, 64) and found that larger segment sizes (32 and 64) provided better accuracy, as they allowed the model to capture a broader context, though they required more computational resources. We also set the number of training epochs to 200, which appeared to strike a balance between sufficient training time and avoiding overfitting. In terms of batch size, we tested sizes 16 and 32, with the larger batch size proving to be more effective due to its more stable gradient updates during training. The Adam optimizer was chosen for its adaptive learning rate, which allowed the model to

converge efficiently. Finally, we found that a learning rate of 0.001 provided the best balance between speed and accuracy, ensuring the model did not converge too quickly or slowly.

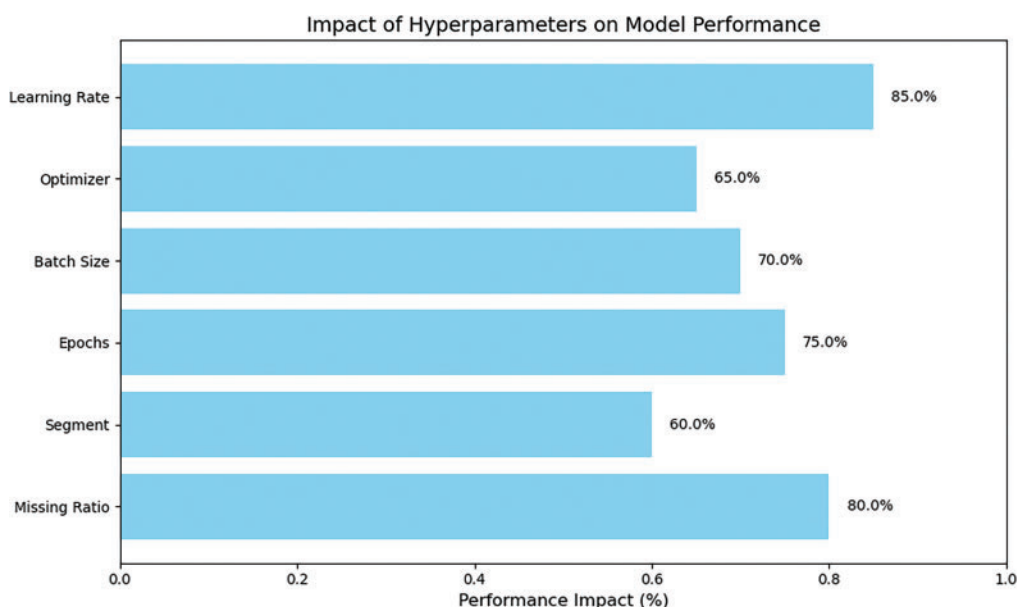


Figure 4: The performance of Dres-CNN model based on hyperparameters

Our proposed DRes-CNN approach shows superior performance compared to other state-of-the-art imputation methods, such as U-Net and LL-CNN. Our proposed validation technique involves splitting the data into 90% for training and validation and 10% for testing. This split allows the model to train on a significant portion of the data while using the separate test set to rigorously evaluate the model's performance on unseen data. Performance evaluation based on the RMSE value shows that the DRes-CNN architecture is more effective in learning features and estimating missing values accurately, even with a significant amount of missing data. Fig. 3, a graphical depiction of Table 4, makes this distinction easy to see. The best value for each dataset is displayed in a column of Table 4 in bold type. As a result, the DRes-CNN architecture produces lower RMSE values compared to the U-Net and LL-CNN architectures. The experimental results also show that in most cases, the DRes-CNN approach outperforms other methods in various data imputations. Although the imputation of missing data with LL-CNN architecture gives better results on the original (clean) version compared to U-Net architecture, DRes-CNN architecture shows better performance on the MIMIC-IV dataset. The superior performance of these imputation methods is most likely due to the suitability of the particular learning algorithm to the data used. Residual connections facilitate deeper network training and utilize convolutional layers with the LeakyReLU activation function to extract features from the input data.

The architecture of DRes-CNN incorporates specialized features like convolutional layers with LeakyReLU activation functions to extract features from input data and have residual connections to facilitate the training of deeper networks. These characteristics render it well-equipped to handle missing or corrupted data within datasets. The ability of DRes-CNN to handle temporal and spatial dependencies within data is very useful *in situations* when missing data appears sequentially, like in time series data. By leveraging convolutional and pooling layers, DRes-CNN effectively captures these dependencies, enhancing predictive accuracy and enabling more effective imputation of missing data values.

Additionally, DRes-CNN considers spatial relationships among features by reorganizing data or applying convolution operations with larger windows to capture broader contexts, facilitating better data imputation. Non-linear information is utilized through non-linear activation functions and optimized training methods, allowing the network to adjust its weights and biases according to the complex data structure. Through optimized training and testing phases, the network is trained on incomplete data to optimize imputation performance and then tested on incomplete data to evaluate its ability to accurately impute missing values.

To see the robustness of the DRes-CNN model, we use box plots to illustrate the impact of handling data. Box plots show the median value, interquartile range, and outliers, giving a clear view of the distribution of the data. Initially, the box plot depicts the data distribution before imputation (Fig. 5a), followed by a representation of the post-imputation process (Fig. 5b to d). The noticeable trend is that the proposed imputation method consistently handles missing data with values to the data mean. Comparing the box plots of the imputed data with the raw data shows a narrower interquartile range and a reduction in data dissemination (Fig. 5d). These observations suggest that the imputation method, rooted in DRes-CNN, effectively minimizes data variability without introducing bias, confirming its suitability for the task at hand.

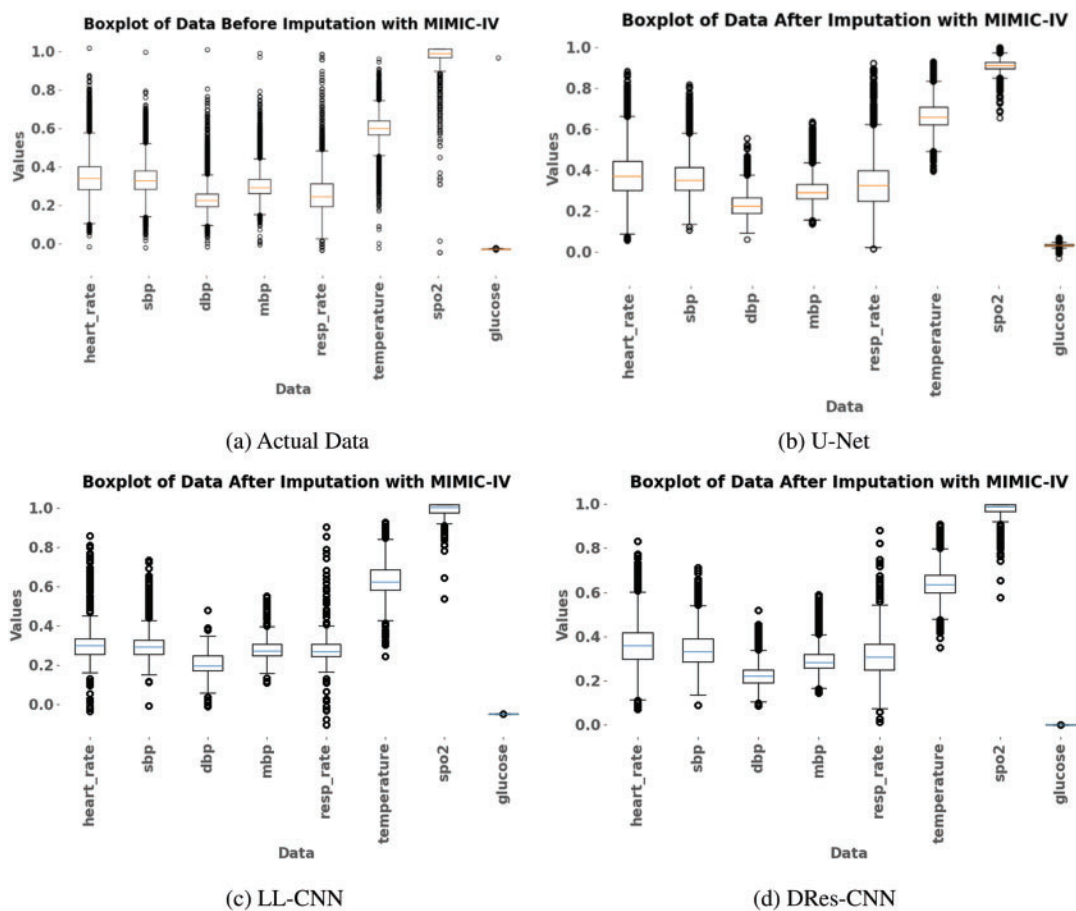


Figure 5: The box plot is utilized for visualizing data representation both pre- and post-imputation using three distinct methods: U-Net, LL-CNN, DRes-CNN

To Validate the robustness of our proposed data imputation model, we conducted a comparative analysis against recent studies that use the existing architecture of the CNN method. However, the use of DRes-CNN architecture for data imputation is still not common. We also use some non-health datasets to test the robustness of our architecture (Table 8). To the best of our knowledge, we are the pioneer in applying the DRes-CNN architecture for data imputation in medical records or other datasets. Our simple DRes-CNN model shows outstanding performance, as evidenced by Table 7, in terms of RMSE beyond previous studies. In the research conducted by Yoon et al. [35], the imputation method used was multi-directional recurrent neural network (M-RNN), which was evaluated on the MIMIC-III database. The evaluation results showed that M-RNN had an RMSE of 0.0295, indicating a fairly good level of imputation accuracy in the context of such medical data.

Table 8: The benchmark of the proposed model with another DL

Reference	Imputation method	Database	MAE	RMSE
Yoon et al. [35]	M-RNN	MIMIC III	–	0.0295
Proposed	U-Net	MIMIC III	0.037	0.0126
		MIMIC-IV	0.007	0.0104
		Beijing Multi-Site Air Quality	0.0087	0.0201
	LL-CNN	MIMIC III	0.00003	0.00045
		MIMIC-IV	0.0351	0.00073
		Beijing Multi-Site Air Quality	0.000162	0.00342
	DRes-CNNs	MIMIC III	0.00000026	0.00018
		MIMIC-IV	0.00000097	0.00006
		Beijing multi-site air quality	0.00000008	0.00002

When it comes to imputing missing data across different datasets, the DRes-CNNs model performs exceptionally well, beating other current techniques such as M-RNN and U-Net50. The DRes-CNNs continue to demonstrate greater accuracy in other databases, including Beijing Multi-Site Air Quality, MIMIC III, and IV, where this trend is also evident. The DRes-CNNs routinely yield the lowest errors even among the suite of suggested models, which includes U-Net and LL-CNN, demonstrating its efficacy in managing challenging data imputation jobs. The DRes-CNNs model's precision and resilience are highlighted by this comparative analysis, which makes it a highly useful tool for situations where exact imputation is crucial.

Despite the proposed DRes-CNN has a good performance. However, to capture the non-linear correlations in the data, for example, the DRes-CNN technique employs learnable weights that are updated iteratively. However, if there are no geographical links in the data, the learning process can proceed incorrectly and produce poor imputation. As is common with most healthcare datasets, our datasets comprised both numerical and categorical factors. It's important to keep in mind, though, that the results of our method could not be the same for data that just includes categorical variables. Imputation may be difficult with this kind of data since there may not be many learnable inter-variable correlations. But solving this problem was outside the purview of our work.

5 Conclusion

In this research, we reviewed and evaluated deep learning (DL) methods for missing value imputation, particularly in healthcare datasets. Our proposed Deep Residual-Convolutional Neural Networks (DRes-CNN) architecture demonstrated significant improvements in imputing missing values over existing methods. The model's performance was assessed using multiple datasets (MIMIC-III, MIMIC-IV, and the Beijing Multi-Site Air Quality), and the results showed that DRes-CNN achieved superior RMSE values compared to other techniques. Despite the proposed DRes-CNN has a good performance. However, to capture the non-linear correlations in the data, for example, the DRes-CNN technique employs learnable weights that are updated iteratively. However, if there are no geographical links in the data, the learning process can proceed incorrectly and produce poor imputation. As is common with most healthcare datasets, our datasets comprised both numerical and categorical factors. It's important to keep in mind, though, that the results of our method could not be the same for data that just includes categorical variables. Imputation may be difficult with this kind of data since there may not be many learnable inter-variable correlations. However, solving this problem was outside the purview of our work. Future work could focus on addressing these limitations by optimizing the architecture for better scalability and reducing computational cost, while also exploring methods to improve the generalization of the DRes-CNN approach to a broader range of datasets, including those dominated by categorical variables. Additionally, expanding the method to handle multimodal data in more complex healthcare scenarios would be another valuable direction for further research.

Acknowledgement: The research infrastructure and financial support for this work was provided by the Intelligent System Research Group (ISysRG), Universitas Sriwijaya, Indonesia.

Funding Statement: This work was supported by the Intelligent System Research Group (ISysRG). This work was also supported by Universitas Sriwijaya funded by the Competitive Research 2024.

Author Contributions: Firdaus Firdaus and Siti Nurmaini: wrote the manuscript and funding acquisition. Anggun Islami and Annisa Darmawahyuni: formal analysis, methodology. Ade Iriani Sapitri and Muhammad Naufal Rachmatullah: designed computer programs, and curated the data. Bambang Tutuko and Akhiar Wista Arum: formal analysis and methodology. Muhammad Irfan Karim and Yultrien Yultrien: resources, analysis, formal analysis, and methodology. Ramadhana Noor Salassa Wandya: contributed data or analysis tools and formal analysis. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets generated and analyzed in this study are available at <https://physionet.org/content/mimiciv/2.2/> (accessed on 23 January 2023).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

- [1] K. Psychogios, L. Ilias, C. Ntanos, and D. Askounis, "Missing value imputation methods for electronic health records," *IEEE Access*, vol. 11, no. 2, pp. 21562–21574, 2023. doi: [10.1109/ACCESS.2023.3251919](https://doi.org/10.1109/ACCESS.2023.3251919).

- [2] M. Y. Landolsi, L. Hlaoua, and L. Ben Romdhane, "Information extraction from electronic medical documents: State of the art and future research directions," *Knowl. Inf. Syst.*, vol. 65, no. 2, pp. 463–516, 2023. doi: [10.1007/s10115-022-01779-1](https://doi.org/10.1007/s10115-022-01779-1).
- [3] M. A. U. Zaman and D. Du, "A stochastic multivariate irregularly sampled time series imputation method for electronic health records," *BioMedInformatics*, vol. 1, no. 3, pp. 166–181, 2021. doi: [10.3390/biomedinformatics1030011](https://doi.org/10.3390/biomedinformatics1030011).
- [4] C. Li *et al.*, "Improving cardiovascular risk prediction through machine learning modelling of irregularly repeated electronic health records," *Eur. Hear. J-Digit. Heal.*, vol. 5, no. 1, pp. 30–40, 2024. doi: [10.1093/ehjdh/zta058](https://doi.org/10.1093/ehjdh/zta058).
- [5] R. H. H. Groenwold, "Informative missingness in electronic health record systems: The curse of knowing," *Diagn. Progn. Res.*, vol. 4, no. 1, pp. 4–9, 2020. doi: [10.1186/s41512-020-00077-0](https://doi.org/10.1186/s41512-020-00077-0).
- [6] K. Yin, D. Qian, and W. K. Cheung, "PATNet: Propensity-adjusted temporal network for joint imputation and prediction using binary EHRs with observation bias," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 6, pp. 2600–2613, 2024. doi: [10.1109/TKDE.2023.3321738](https://doi.org/10.1109/TKDE.2023.3321738).
- [7] H. Khan, X. Wang, and H. Liu, "Missing value imputation through shorter interval selection driven by Fuzzy C-Means clustering," *Comput. Electr. Eng.*, vol. 93, no. 2, pp. 1–16, 2021. doi: [10.1016/j.compeleceng.2021.107230](https://doi.org/10.1016/j.compeleceng.2021.107230).
- [8] A. D. Woods *et al.*, "Best practices for addressing missing data through multiple imputation," *Infant. Child Dev.*, vol. 33, no. 1, pp. 1–37, 2024. doi: [10.1002/icd.2407](https://doi.org/10.1002/icd.2407).
- [9] L. O. Joel, W. Doorsamy, and B. S. Paul, "On the performance of imputation techniques for missing values on healthcare datasets," 2024, *arXiv:2403.14687*.
- [10] Q. Suo, W. Zhong, G. Xun, J. Sun, C. Chen and A. Zhang, "GLIMA: Global and local time series imputation with multi-directional attention learning," in *Proc. 2020 IEEE Int. Conf. Big Data (Big Data)*, Atlanta, GA, USA, 2020, pp. 798–807. doi: [10.1109/BigData50022.2020.9378408](https://doi.org/10.1109/BigData50022.2020.9378408).
- [11] P. C. Austin, I. R. White, D. S. Lee, and S. van Buuren, "Missing data in clinical research: A tutorial on multiple imputation," *Can J. Cardiol.*, vol. 37, no. 9, pp. 1322–1331, 2021. doi: [10.1016/j.cjca.2020.11.010](https://doi.org/10.1016/j.cjca.2020.11.010).
- [12] X. Weng *et al.*, "A joint learning method for incomplete and imbalanced data in electronic health record based on generative adversarial networks," *Comput. Biol. Med.*, vol. 168, 2024, Art. no. 107687. doi: [10.1016/j.compbiomed.2023.107687](https://doi.org/10.1016/j.compbiomed.2023.107687).
- [13] J. Wang *et al.*, "Deep learning for multivariate time series imputation: A survey," 2024, *arXiv:2402.04059*.
- [14] S. Hong and H. S. Lynn, "Accuracy of random-forest-based imputation of missing data in the presence of non-normality, non-linearity, and interaction," *BMC Med. Res. Methodol.*, vol. 20, no. 1, pp. 1–13, 2020. doi: [10.1186/s12874-020-01080-1](https://doi.org/10.1186/s12874-020-01080-1).
- [15] L. Cappelletti, T. Fontana, G. W. Di Donato, L. Di Tucci, E. Casiraghi and G. Valentini, "Complex data imputation by auto-encoders and convolutional neural networks—A case study on genome gap-filling," *Computers*, vol. 9, no. 2, pp. 37, 2020. doi: [10.3390/computers9020037](https://doi.org/10.3390/computers9020037).
- [16] E. Macias Toro, G. Boquet, J. Serrano, J. Lopez Vicario, J. Ibeas and A. Morell, "Novel imputing method and deep learning techniques for early prediction of sepsis in intensive care units," in *2019 Comput. Cardiol. Conf. (CinC)*, Singapore, 2019, vol. 45, pp. 1–4. doi: [10.22489/cinc.2019.038](https://doi.org/10.22489/cinc.2019.038).
- [17] D. A. Neu, J. Lahann, and P. Fettke, "A systematic literature review on state-of-the-art deep learning methods for process prediction," *Artif. Intell. Rev.*, vol. 55, no. 2, pp. 801–827, 2022. doi: [10.1007/s10462-021-09960-8](https://doi.org/10.1007/s10462-021-09960-8).
- [18] H. Khan, X. Wang, and H. Liu, "Handling missing data through deep convolutional neural network," *Inf. Sci.*, vol. 595, no. 5–6, pp. 278–293, 2022. doi: [10.1016/j.ins.2022.02.051](https://doi.org/10.1016/j.ins.2022.02.051).
- [19] A. D. Kulkarni, "Fuzzy convolution neural networks for tabular data classification," *IEEE Access*, vol. 12, pp. 151846–151855, 2024. doi: [10.1109/ACCESS.2024.3479882](https://doi.org/10.1109/ACCESS.2024.3479882).
- [20] M. I. Iqbal, M. S. H. Mukta, A. R. Hasan, and S. Islam, "A dynamic weighted tabular method for convolutional neural networks," *IEEE Access*, vol. 10, no. 1, pp. 134183–134198, 2022. doi: [10.1109/ACCESS.2022.3231102](https://doi.org/10.1109/ACCESS.2022.3231102).

- [21] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, 2021. doi: [10.1109/TNNLS.2021.3084827](https://doi.org/10.1109/TNNLS.2021.3084827).
- [22] Y. Wen, K. Yi, J. Ke, and Y. Shen, "DiffImpute: Tabular data imputation with denoising diffusion probabilistic model," 2024, *arXiv:2403.13863*.
- [23] C. C. Nguyen, "Deep learning for simultaneous imputation and classification of time series incomplete data," *J. Sci. Tech.*, vol. 12, no. 1, pp. 110–125, 2023. doi: [10.56651/lqdtu.jst.v12.n1.661.ict](https://doi.org/10.56651/lqdtu.jst.v12.n1.661.ict).
- [24] X. Lai, X. Wu, and L. Zhang, "Autoencoder-based multi-task learning for imputation and classification of incomplete data," *Appl. Soft Comput.*, vol. 98, 2021, Art. no. 106838. doi: [10.1016/j.asoc.2020.106838](https://doi.org/10.1016/j.asoc.2020.106838).
- [25] Z. Dai, E. Getzen, and Q. Long, "SADI: Similarity-aware diffusion model-based imputation for incomplete temporal EHR data," in *Proc. 27th Int. Conf. Artif. Intell. Statist.*, València, Spain, 2024, pp. 4195–4203.
- [26] D. Varma, C. S. Yainik, A. Thorave, and N. Sharma, "Handling missing data in longitudinal anthropometric data using multiple imputation method," in *Int. Conf. Data Manag., Analyt. Innov. (ICDMAI 2024). Lect. Not. Netw. Syst.*, Vellore, India, 2024, vol. 997, pp. 273–287.
- [27] J. Li *et al.*, "Method for incomplete and imbalanced data based on multivariate imputation by chained equations and ensemble learning," *IEEE J. Biomed. Health Inform.*, vol. 28, no. 5, pp. 3102–3113, 2024. doi: [10.1109/JBHI.2024.3376428](https://doi.org/10.1109/JBHI.2024.3376428).
- [28] P. Brimos, P. Seregkos, A. Karamanou, E. Kalampokis, and K. Tarabanis, "Deep learning missing value imputation on traffic data using self-attention and GAN-based methods," in *2024 Panhellenic Conf. Elect. Telecommun. (PACET)*, 2024, pp. 1–4.
- [29] Y. Yin, Z. Yuan, I. M. Tanvir, and X. Bao, "Electronic medical records imputation by temporal generative adversarial network," *BioData Min.*, vol. 17, no. 1, p. 19, 2024. doi: [10.1186/s13040-024-00372-2](https://doi.org/10.1186/s13040-024-00372-2).
- [30] H. Khan, M. T. Rasheed, H. Liu, and S. Zhang, "High-order polynomial interpolation with CNN: A robust approach for missing data imputation," *Comput. Electr. Eng.*, vol. 119, 2024, Art. no. 109524. doi: [10.1016/j.compeleceng.2024.109524](https://doi.org/10.1016/j.compeleceng.2024.109524).
- [31] A. Johnson, L. Bulgarelli, T. Pollard, S. Horng, L. A. Celi and R. Mark, "MIMIC-IV," *PhysioNet*. 2020. Accessed: Jan. 23, 2023. [Online]. Available: <https://physionet.org/content/mimiciv/1.0/>
- [32] A. E. W. Johnson *et al.*, "MIMIC-IV, a freely accessible electronic health record dataset," *Sci. Data.*, vol. 10, no. 1, pp. 1–10, 2023. doi: [10.1038/s41597-022-01899-x](https://doi.org/10.1038/s41597-022-01899-x).
- [33] C. Ghosh, "Data pre-processing," in *Data Analysis with Machine Learning for Psychologists: Crash Course to Learn Python 3 and Machine Learning in 10 Hours*. Berlin, Heidelberg, Germany: Springer, 2022, pp. 55–85.
- [34] I. Izonin, R. Tkachenko, N. Shakhovska, B. Ilchyshyn, and K. K. Singh, "A two-step data normalization approach for improving classification accuracy in the medical diagnosis domain," *Mathematics*, vol. 10, no. 11, pp. 1–18, 2022. doi: [10.3390/math10111942](https://doi.org/10.3390/math10111942).
- [35] J. Yoon, W. R. Zame, and M. van der Schaar, "Estimating missing data in temporal data streams using multi-directional recurrent neural networks," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 5, pp. 1477–1490, 2019. doi: [10.1109/TBME.2018.2874712](https://doi.org/10.1109/TBME.2018.2874712).