IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.0092316

Missing value imputation methods for electronic health records

KONSTANTINOS PSYCHOGYIOS[1], LOUKAS ILIAS[1], CHRISTOS NTANOS[1], and DIMITRIS ASKOUNIS[1]

[1]Decision Support Systems Laboratory School of Electrical and Computer Engineering National Technical University of Athens 15780 Athens, Greece

Corresponding author: Konstantinos Psychogyios (e-mail: kwstaspsychogios@gmail.com).

This research has been funded by the European Union through the Horizon 2020 Research and Innovation Programme, in the context of the MES-CoBraD (Multidisciplinary Expert System for the Assessment & Management of Complex Brain Disorders) project under grant agreement No GA 965422.

ABSTRACT Electronic health records (EHR) are patient-level information, e.g., laboratory tests and questionnaires, stored in electronic format. Compared to physical records, the EHR alternative allows patients to access their data easily and helps staff with management procedural tasks such as information sharing across different organizations. Moreover, this type of data is commonly used by researchers for predictive and classification purposes, employing statistical and machine learning methods. However, missingness is a phenomenon that is observed very frequently for such measurements. Even though this missingness is often significant, it is usually treated poorly with either case deletion or simple methods, resulting in suboptimal and/or inaccurate predictive results. This happens because the simple methods, e.g., k-nearest neighbors (kNN) and mean/mode imputation, fail in most cases to incorporate the complex relationships that define these medical datasets. To address these limitations, in this paper we test and improve state-of-the-art missing data imputation models and practices. We propose a new missing value imputation method based on denoising autoencoders (DAE) with kNN for the pre-imputation task. We optimize the training methodology by re-applying kNN to the missing data every N epochs using a different value for the variable k each time to yield more accurate results. We also revise a state-of-the-art missing data imputation approach based on a generative adversarial network (GAN). Using this as a baseline, we introduce improvements regarding both the architecture and the training procedure. These models are compared with the ones usually employed within clinical research studies for both the task of imputation and post-imputation prediction. Results show that our proposed deep learning approaches outperform the standard baselines, yielding better imputation and predictive results.

INDEX TERMS Missing value imputation, Deep learning, Generative Adversarial Networks, Autoencoders, missing data, EHR.

I. INTRODUCTION

An electronic health record (EHR) is a document that contains medical information, e.g., laboratory measurements, for a patient and is stored online. Thus, it can be shared across multiple facilities and accessed quickly by patients or medical staff. An EHR is used primarily for purposes of setting objectives and planning patient care, documenting the delivery of care, and assessing the outcomes of that care [1]. These data provide opportunities to enhance patient care, embed performance measures in clinical practice, and facilitate clinical research [2]. An example of research based on EHRs is the prediction of cardiovascular risk using machine learning regression methods [3]. Such a model can be used as a decision-support system to help doctors and physicians manage patients and act proactively.

However, it is very common for this type of data to have a percentage of missingness [4]. Missing data occurs when the values of the variables of interest are not measured or recorded for all subjects in the sample. Data can be missing for several reasons [5], including: (i) patient refusal to respond to specific questions, e.g., patient does not report data on income; (ii) loss of patient to follow-up; (iii) investigator or mechanical error, e.g., sphygmomanometer failure; and (iv) physicians not ordering certain investigations for some patients, e.g., cholesterol test not ordered for some patients. The missing values can be defined by three main mechanisms:

(a) missing completely at random (MCAR), (b)missing at random (MAR) and (c) missing not at random (MNAR) [6]. The first case (MCAR) occurs when the missingness presented in an EHR follows a totally random pattern. For example, if we have EHR lab measurements regarding cardiovascular disease, e.g., blood pressure and cholesterol, some patients may have missing values at cholesterol because they were not able to visit the lab this specific day due to a public transport strike. The second (MAR) indicates that missingness in one variable is related to an other variable. An example could be that missing data on diastolic blood pressure is related to low systolic blood pressure. The third case (MNAR) indicates that missingness in a variable is dependent on the variable itself. An instance of such a case is when people with high cholesterol don't visit the hospital to perform lab tests. It is also worth noting that within a clinical EHR dataset more than one missing pattern may be present at the time, with varying percentages of missingness. Within the clinical research framework, missing data are usually handled poorly [7]. The most common approach is the complete case analysis where rows containing missing values either at the predictor or the outcome variables are dropped. This choice is highly problematic since it leads to a smaller dataset and a model that is not able to generalize well. Also, this method frequently produces results and errors that may be small for the complete subset of data but in reality are optimistic. Moreover, different studies may use different subsets of the same dataset, e.g., instead of rows, columns may be dropped or a combination of both, and this choice makes comparisons harder. Another approach to solving this issue is simple imputation with the mean, mode (most frequent) algorithm, or kNN imputer [8]. These lead to a complete dataset but are too simple and thus impute values that are unrealistic. For instance, concerning cardiovascular disease (CVD) patient-level data, there is usually a strong correlation between the corresponding variables, e.g., systolic and diastolic blood pressure, which should be incorporated into the missing value imputation model. This is something that univariate statistical approaches and simple regression algorithms fail to compute, leading to inaccurate results [9]. These correlations are of course present in most medical datasets where tests have been conducted for the same patient, lab measurements have been carried out for a specific task etc. Some somewhat more complex and better-resulting methods are Missforest (MF) [10] and multivariate imputation by chained equations (MICE) [11]. Even though these methods are more sophisticated, they still lack the capacity to fully analyze the complex relationships that define EHR datasets [12], [13]. This is something that is more severe in longitudinal studies where information regarding a missing value should be correlated with previous values for the same patient.

To address these limitations, in this paper we compare several methods of missing data imputation. Specifically, we propose two deep learning approaches based on the denoising autoencoder (DAE) and generative adversarial networks (GANs). Motivated by [14] we propose a method based on a DAE using kNN for pre-imputation. Using this model as a baseline, we implement various changes regarding both the architecture and the training process, which yield considerably more accurate results. We customize the loss function for mixed types of datasets (categorical and numerical) and add batch normalization. Also instead of using kNN once at the start of the training we re-apply it every N rounds with a different value for k. Furthermore, the training procedure also includes the artificial introduction of missingness to a complete dataset for the autoencoder to impute. Instead of doing this once at the start of the training process, we implement it at the end of every epoch, changing the location of the variables that are to be imputed. These changes contribute to the kNN-DAE approach by improving the training methodology in a way that allows broader learning. In terms of the GAN approach, we also build upon the existing architecture, making improvements concerning the specific case we study. More specifically, we use a DAE with kNN pre-imputation as the generator and apply the aforementioned adjustments to the training procedure. We also compare our approaches with traditional ones (e.g., mean, mode, and plain kNN) and show that our approaches yield considerable advantage. To assess our models, we use four publicly available EHR datasets. Finally, the proposed models are evaluated for both the imputation and post-imputation prediction tasks. We study the latter to explore whether the choice of more robust imputation methods will result in higher predictive performance or not. This is very important since prediction is usually the common goal of researchers and practitioners when applying machine learning techniques to EHR data.

Our main contributions can be summarized as follows:

- We present a new missing value imputation method based on a DAE with kNN pre-imputation.
- We extend and improve a state-of-the-art missing value imputation method based on GAN.
- We evaluate our methods with four publicly available EHR datasets and introduce different types of missingness to these datasets to account for all cases that could be present in a real-world scenario.
- We evaluate our methods for both the task of imputation and post-imputation prediction and show that our introduced approaches demonstrate valuable advantages over the state-of-the-art ones.

The current manuscript is an extension to the previous work presented in the 2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI) [15] and introduces more comprehensive analyses and evaluations, including new improvements on the deep learning models, more EHR datasets, and additional commonly used methods to compare with.

II. RELATED WORK

Approaches towards missing value imputation employing EHRs data vary in the literature. There is not a single solution

that data, etc.fits all cases, and usually researchers select algorithms that perform better for the specific task. The key reasons behind choosing an imputation technique are the mechanism of missingness, the gap length of missing data etc. [16].

The simplest way of dealing with this is by deleting the records that contain at least one missing value and thus selecting a subset of the original dataset where there are no missing values [17]–[20]. Gupta et al. [21] studied the case of obesity prediction with EHR data using common machine learning models such as random forests and LSTMs. When it comes to missing values, they dropped rows with missing or corrupt values, e.g., implausible dates. At the same time, they dropped columns where more than 50% of the entries were corrupt. Kwakye and Dadzie [22] studied the case of coronary heart disease using the Framingham Heart Study dataset, which is available at Kaggle. Regarding the pre-processing steps, they chose to eliminate both missing and outlier data, resulting in an undercomplete dataset.

Another popular but simple approach is the imputation based on mean, mode or zero imputation where each missing value is imputed by zero [23]. Liu et al. [24] exploited a dataset of clinical trials (possibly many per patient) and adopted zero imputation for the result of a test if the value was missing and it was the first time the particular test was conducted. If there has been a past clinical trial involving this patient in which this test was performed, the corresponding outcome was used to fill the missing value.

Guo et al. [25] developed a deep learning approach for the problem of heart failure prediction using synthetic EHR data. In their analysis, they chose to discard features with more than 50% of the entries missing and impute the rest using the mean and most frequent values for numerical and categorical features, respectively. Gupta et al. [26] evaluated machine learning models for the problem of heart attack prediction using the Framingham Heart Study dataset and the Heart dataset for the UCI Machine Learning Repository. When it comes to pre-processing and especially missing values, their approach was to impute using the mean or median, where the latter was preferred for features with skewed distributions. Concerning categorical missingness, it was dealt with the addition of an extra category for "missing." kNN and MLP are two methods that have also been used to address this issue. Jerez et al. [27] applied missing data imputation methods to a real world breast cancer dataset with an overall 5.61% percent of missingness. They utilized kNN, MLP, MICE, SOM, etc. algorithms for this problem. They found that the best performing method was kNN, leading to higher post-imputation accuracy.

Recent advances in deep learning have produced state-of-the-art results by modifying existing models to fit the missing value imputation framework. These advances can be categorized as either discriminative or generative. Yoon et al. [28] modified the original GAN architecture and created a generative adversarial imputation network (GAIN). Results showed that this approach surpasses robust imputation methods, including autoencoder-based methods. Dong et al. [29] evaluated modern missing value imputation methods such as GAIN, MICE, and Missforest. They employed two real-world datasets to support their claims. Results show that the deep learning approach achieves better performance, and this is something that is more obvious when the missing percentage is high. Park et al. [30] gathered EHR data from wearable devices with the intent of prediction based on machine learning. In this dataset, the proportion of missing data was 2.83%. To impute this, they first evaluated missing value imputation methods on a complete subset of the data. The methods were: (i) GAIN, (ii) kNN, (iii) mean, mode. Results show that the deep learning approach outperforms the simpler methods by a substantial margin.

Regarding discriminative models, Aidos and Tomás [14] proposed an overcomplete DAE with kNN pre-imputation. This approach was tested against many common missing value imputation methods such as MICE, GAIN, Least square imputation and mean, mode. Findings suggest that these methods outperform the standard ones and can handle high missing rates of up to 50%. Seo et al. [31] tested a denoising autoencoder with kNN pre-imputation for gas data. The comparison was done against common missing value imputation approaches. Results showed that the autoencoder solution achieved the best performance.

III. PROBLEM FORMULATION

A. MISSING VALUE IMPUTATION

The missing value imputation problem results if features of a dataset have unobserved values. Consider a random variable $X = (X_1, X_2, .., X_N) \in X^N$ where X represents the space to which each sample belongs to and has a distribution of $P(X)$. Consider also a mask vector $M = (M_1, M_2, .., M_N)$ where each M_i takes values in $\{0, 1\}$ and $M_i = 1$ means the value is observed opposing to $M_i = 0$ which means the value is missing. Having d instances of X and M we define a dataset (X^i, M^i) for $i = 0, 1, .., d$. From this, $(\hat{X}^i, M^i)$ is derived substituting each feature j connected to a sample i if $M_{i,j} = 0$ with a pre-imputation value (possibly random noise). In such a case, given a model IMP our goal is to create an imputed dataset $\tilde{X}^i = IMP(\hat{X}^i, M^i)$ for $i = 0, 1, .., d$. Each imputed sample should be generated based on $P(X|\tilde{X} = \tilde{X}^i)$ since we want our imputed data to follow the original dataset's distribution.

The result is a new dataset $\bar{X}$ where for each sample i we have :

$$\bar{X}^i = X^i \odot M^i + (1 - M^i) \odot \tilde{X}^i \tag{1}$$

B. POST-IMPUTATION PREDICTION

Applying different missing value imputation algorithms A_i for $i = 0, 1, ..S$ to the original dataset X results in S different datasets $\bar{D}_1, \bar{D}_2, .., \bar{D}_S$. For each of these datasets, we use a standard predictive method P to predict the patient's heart disease status (binary classification).

IV. METHODS

In this section, we present our proposed approaches. Specifically, we employ various missing value imputation methods, evaluating them on EHR datasets. We use simple statistical approaches and multiple imputation techniques, as well as more complex ones such as GAN and DAE. Below, we describe in detail the introduced approaches.

A. SIMPLE

This model is the simple statistical approach of mode, mean imputation. We impute categorical missing values using the most frequent class and numerical variables using the mean obtained by the corresponding column.

B. KNN

We employ kNN, which imputes missing data considering the distance between the sample vectors in the dataset's space. For each feature missing, it considers the k closest samples that have this feature observed and averages their values regarding numerical data. When it comes to categorical data, the result is the most frequent class of the k nearest neighbors. In our case for $k = 5$, given a sample $S(X, Y, 0)$ and its 5 nearest neighbors $N_5 = \{(X_i, Y_i, 1)|i = 1, 2, .., 5\ \}$ we define:

$$Y = \begin{cases} argmax_z\{\sum_{(X_iY_i,1)\in N_5)} 1(Y_i = z)\} & \text{if } Y \text{ is categorical} \\ \frac{1}{5}\sum_{i=1}^{5} Y_i & \text{if } Y \text{ is numerical} \end{cases} \tag{2}$$

where z can be either 0 or 1 since we only have binary values, and $1(Y_i = z)$ is a function that returns 0 if $(Y_i = z)$ and otherwise returns 0. The metric to measure the distance between two points p and q is the euclidean:

$$d(p, q) = \sum_{i=1}^{n}(q_i - p_i)^2 \tag{3}$$

where n is the number of variables for each data point.

C. MISSFOREST

We exploit the method proposed by Stekhoven Daniel J. and Bühlmann Peter [32]. In this model, firstly, all columns except one are imputed using mean and mode imputation. Consequently, a random forest is used to predict the missing values in the column that was excluded before. The predicted values are then used as imputations. This process iterates through the data in a loop, where each iteration builds upon the last, improving the imputed variables. The procedure continuous until the difference between the imputed matrix M_{new}^{imp} and M_{old}^{imp} is not increasing. In our case we also use max iterations = 20 and number of trees = 100. The aforementioned difference for numerical features N is :

$$\delta_N = \frac{\sum_{j\in N}(M_{new}^{imp} - M_{old}^{imp})^2}{\sum_{j\in N}(M_{new}^{imp})^2} \tag{4}$$

And for the categorical F:

$$\delta_F = \frac{\sum_{j\in F}\sum_{i=1}^{i=n} I_{M_{new}^{imp} \neq M_{old}^{imp}}}{F_{NA}} \tag{5}$$

where F_{NA} is the number of missing values in categorical variables.

D. MICE

Multivariate Imputation by Chained Equations was introduced by van Buuren et al. [33]. This method consists of three main steps, namely imputation, analysis, and the pooling step. Regarding the first, data are imputed n times, resulting in n different datasets. Each dataset is handled separately in the analysis step to obtain parameter estimates and standard errors. Finally, all copies are pooled together to obtain the overall results. For example, regarding a parameter P_i from the i^{th} dataset we have an overall estimate of:

$$\bar{P} = \frac{1}{n}\sum_{i=1} nP_i \tag{6}$$

and overall variance:

$$\bar{V} = \frac{1}{n}\sum_{i=1} nV_i \tag{7}$$

There is also the variability of imputed values between different(n) datasets. This variance is denoted as:

$$\bar{B} = \frac{1}{n-1}\sum_{i=1} n(P_i - \bar{P})^2 \tag{8}$$

E. NEIGHBORHOOD AWARE AUTOENCODER (NAA)

Autoencoders in general consist of an encoder and a decoder part that recreate the input on the output level. These models are commonly used for feature extraction and selection. Denoising autoencoders constitute a variation of this approach, where the input is corrupted. Since a dataset with missing values can be considered corrupt, denoising autoencoders are commonly used for missing value imputation, as mentioned in the Related Work section. The encoder maps an input vector $\vec{X}$ to a hidden layer $\vec{Y}$ with the application of a nonlinear transformation:

$$f_{enc}(\vec{X}) = s(\vec{X} \cdot W^T + b) \tag{9}$$

where $\vec{Y} = f_{enc}(\vec{X})$, W is the weight matrix and b the bias vector of the encoder. The $\vec{Y}$ hidden vector constituting the result of this process is consequently mapped by the decoder to the output which has the same dimension as $\vec{X}$. The output $\vec{Z}$ thus is the nonlinear transformation:

$$f_{dec}(\vec{Y}) = s(\vec{Y} \cdot \bar{W}^T + \bar{b}) \tag{10}$$

where $\vec{Z} = f_{dec}(\vec{Y})$, $\bar{W}$ is the weight matrix and $\bar{b}$ the bias vector of the encoder. The weights and biases of both the encoder and decoder are trainable parameters, and the purpose is to find the corresponding values that minimize the reconstruction error between $\vec{X}$ and $\vec{Z}$.
Based on this, we utilize the autoencoder model proposed by

Aidos and Tom [14], named neighborhood-aware autoencoder (NAA). In this work, an overcomplete DAE is used for the problem of missing value imputation, where the pre-imputation is done with kNN. Jointly these two formulate the kNN-DAE approach. The pre-imputation part is conducted for the whole dataset before model training using $k = 5$. By using kNN as a pre-imputation method, it is harder to converge to local minimums during the early epochs of training since the initial estimation is decent.

F. IMPROVED NEIGHBORHOOD AWARE AUTOENCODER (I-NAA)

Based on the NAA approach, we make some improvements. Firstly, we empirically choose an undercomplete architecture with half the input size for the encoder's output.

Regarding the training process of autoencoders, consider a copy D_{copy} of the original complete dataset D. First we introduce missingness to D_{copy} and then replace these missing values with noise or in our case with kNN imputation resulting in $D_{imputed}$. Then batches of $D_{imputed}$ are fed iteratively to the network which must learn to map them to the corresponding batches of D. With constraints applied to the network, the result of this process is that the autoencoder learns information about the relationships between the features.

However, if for the whole training process the train pre-imputed values of $D_{imputed}$ remain the same, the model may learn a mapping from these specific values to the actual ones and not the relationships between the features that define the dataset. Since our goal is both the latter and consideration of the local neighborhood, we change the pre-imputed values every *N* epochs, where *N* is empirically chosen. Specifically, *N* is equal to 10. Every 10 epochs, we still use kNN for pre-imputation, but we change the value of k (closest neighbors) to one that hasn't been used before and is within certain $[B1, B2]$ bounds. In the same manner, if the values that are to be imputed, e.g. X_{12} and X_{23}, remain the same for the whole training procedure, the autoencoder may learn to impute only these specific values. For this reason, we change the values that are to be imputed at the start of each epoch. This is an additional constraint applied to our model that prevents convergence to a local minimum. We also create a custom loss function tailored to our dataset. Multi-label features are one-hot encoded. Considering N numerical and C binary categorical features, we rearrange the features in the dataset so as to ensure that the numerical features are in the first N column indexes and the categorical are in indexes of $(N+1, N+C)$. In terms of the numerical variables, we minimize the mean squared error (MSE), as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i - Y_i)^2 \tag{11}$$

Regarding the categorical features, we use the binary cross entropy (BCE):

$$BCE = -\frac{1}{N}\sum_{i=N+1}^{N+C} y_i \cdot log(p(y_i) + (1-y_i) \cdot log(1-p(1-y_i)) \tag{12}$$

so jointly:

$$Loss = RMSE + BCE \tag{13}$$

This architectural and training approach is illustrated in Fig. 1.

G. GAIN

This model was proposed by Yoon, J et al. [28] and is based on the original GAN architecture. GAIN adopts the generator (G) and discriminator (D) architecture, but now the discriminator, instead of classifying the whole output of the generator as true or false, categorizes each variable from the vector emitted as imputed or real. When the generator imputes missing values that resemble the real distribution, the discriminator can no longer discriminate real from fake, and the model has converged. The architecture proposed also uses a hint mechanism where, in addition to the generator's output, the discriminator is given a hint containing information about the missing values. The hint H is dependent on M and basically is proportionally identical to it. For example, if this proportion is set to 90%, nine out of ten variables in the hint vector have the same value as those in the mask vector. The generator's job in such a case is to find the correct values for the remaining 10%. In the original paper it is proved that if the hint is not large enough there are several distributions that G could reproduce that would all be optimal with respect to D.

More specifically the generator given an random vector Z outputs an imputed dataset $\bar{X}_i$ from which we obtain $\bar{X}_i$ based on equation 1. Based on this, we define:

$$L_D(M, \bar{M}, H) = \sum_{i:H_i=0} [M_i \cdot log(\bar{M}_i + (1-M_i) \cdot log(1-M_i)]$$

where M is the true mask vector ,$\bar{M}$ the generated mask vector and H is the hint vector. It is noted that the sum is calculated only on values where $H_i = 0$ because otherwise the discriminator would overfit to the hint vector. Thus, the discriminator is trained based on:

$$min_D - \sum_{i=1}^{batchsize} L_D(M_i, \bar{M}_i, H_i) \tag{14}$$

Regarding the generator we define:

$$L_G(M, \bar{M}, H) = -\sum_{i:H_i=0} (1 - M_i) \cdot log(M_i) \tag{15}$$

which intuitively is a value that measures how often the generator fools the discriminator. Secondly, we define:

$$L_M(\tilde{X}, \bar{X}) = -\sum_{i=1}^{d} M_i \cdot Diff(\tilde{X}_i, \bar{X}_i) \tag{16}$$

where d is the dimension size of the dataset and :

$$Diff(\tilde{X}, \bar{X}) \begin{cases} (\tilde{X}_i - \bar{X}_i)^2 & \text{if } X_i \text{ is numerical} \\ -X_i log(\bar{X}_i) & \text{if } X_i \text{ is binary} \end{cases} \tag{17}$$

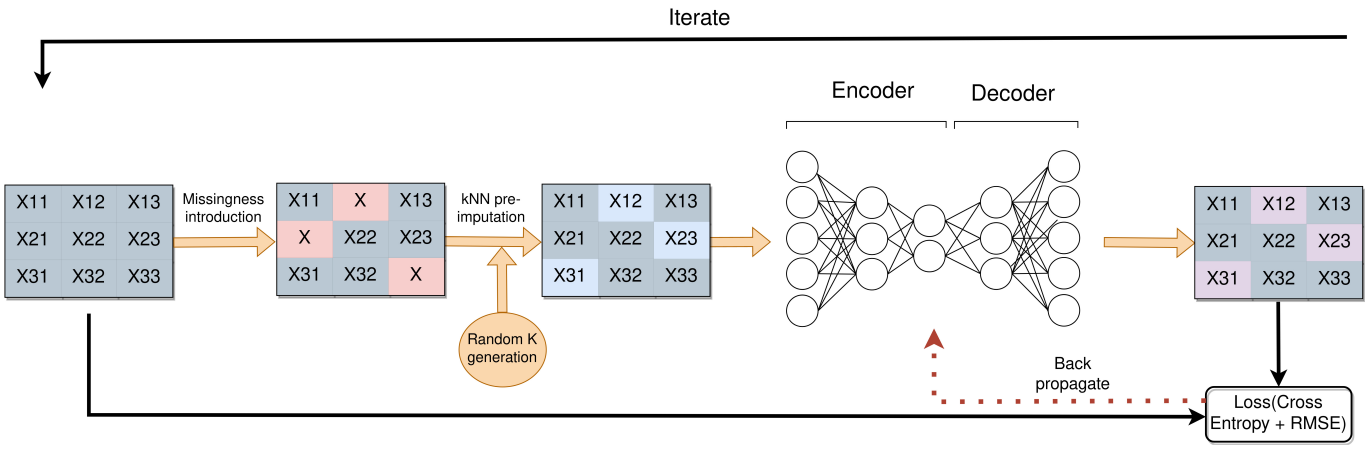

FIGURE 1: I-NAA training methodology.

Equation 16 on the other hand measures how accurate is the generator in recreating the observed components of the input. At last, the generator is trained to minimize the equation defined below:

$$min_G \sum_{i=1}^{batchsize} L_G(M_i, \bar{M}_i, H_i) + \alpha L_{M_i}(\tilde{X}_i, \bar{X}_i) \quad (18)$$

where α is a scaling parameter.

H. IMPROVED GENERATIVE ADVERSARIAL IMPUTATION NETWORK (I-GAIN)

Based on the GAIN approach, we propose some changes. Firstly, we add batch normalization both to the generator and the discriminator. Secondly, similar to the case of the autoencoder, we pre-impute with the kNN algorithm using different values of k (the number of neighbors) every N epochs instead of random noise. We also utilize the same custom loss function as described in ***I-NAA***. In addition, the authors in [28] proposed a simple 3 layer architecture for the Generator part, where each layer had the same number of units. We replace this structure with one with 5 dense layers in an undercomplete autoencoder architecture since this model has proven to be effective in learning the dataset's distribution. It can be thus noted that the training methodology and architecture of I-NAA have been incorporated into the GAIN algorithm. Moreover, GAN training is known to be difficult with problems such as model collapse. To tackle this limitation, we add a gradient penalty to the discriminator's loss as defined by Gulrajani et al. [34]:

$$GP = \frac{a}{batchsize} \sum_{i=1}^{batchsize} (\|\nabla \bar{X}_i \cdot D(\bar{X}_i)\|_2 - 1)^2 \quad (19)$$

where α is a scaling parameter. After adding the gradient penalty to the loss of the discriminator we experimented with (a) original generator as proposed by [28] & (b) our introduced modified generator. Findings showed that the usage of the modified generator yielded better results than the original one.

This architectural and training approach is illustrated in Fig. 2. Here, we can view a separation of the training procedure into two subprocesses. Firstly, the data flows through the network, which can be seen by the yellow arrows. Secondly, based on the outputs of these subprocesses, namely the output of the generator and the output of the discriminator, the losses are calculated and are subsequently back propagated into the parameters of the neural networks.

V. DATASETS

To assess the performance of our models, we employ four EHR datasets. Three of these four datasets contain missing values with various percentages of missingness. These are real-world datasets, meaning that feature correlations are present and strong. They are also datasets that have been used in the literature to evaluate missing data imputation methods, so direct comparisons with other works are possible.

A. FRAMINGHAM HEART STUDY

We use the publicly available Framingham heart dataset [1]. The Framingham Heart Study is a longitudinal cardiovascular cohort study consisting of medical, laboratory, and questionnaire events for 4.434 participants. Originally, the dataset consisted of 39 variables, but we only used 15 for the purpose of this study. Examples of such variables are cigarettes per day and body mass index (BMI). This subset has 8 numerical and 7 categorical where the latter are all binary. Our target variable is coronary heart disease, which is binary. The dataset is also not complete, with feature missingness varying from 0 to 13%.

[1] https://biolincc.nhlbi.nih.gov/studies/framcohort/

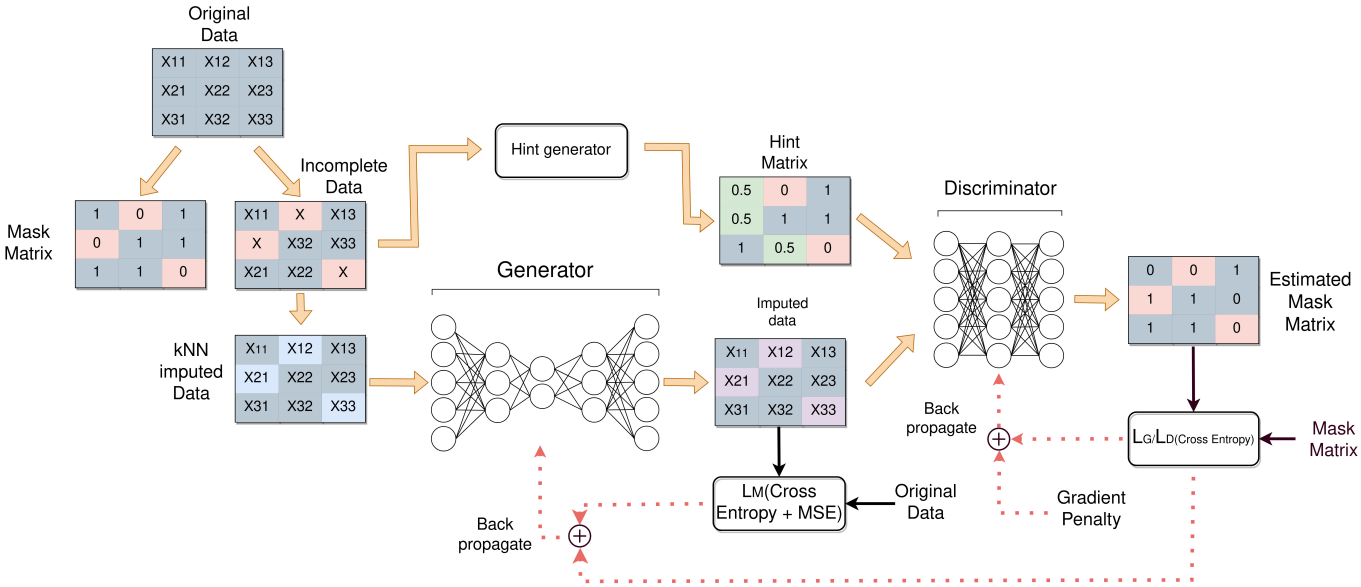

FIGURE 2: I-GAIN training methodology.

B. STROKE DATASET

This is a publicly available stroke prediction dataset from Bangladesh, accessible at Kaggle[2]. The dataset has 5110 samples and twelve attributes, which are BMI, average glucose level, etc. Three of the variables are numerical, whereas the remaining nine are categorical. It is also noted that the dataset is highly imbalanced, with 249 positive cases and 4861 negative. Moreover, BMI is the only feature with missing values with a proportion of 3.933 %. Our target variable is stroke, which is binary (True or False).

C. PHYSIONET HEART FAILURE

This is an EHR dataset of hospitalized patients with heart failure [35]. The dataset consists of 167 variables, both numerical and categorical. Since we want to evaluate whether a missing value imputation model can understand complex correlations in medical datasets, we only kept numerical variables that are highly correlated. Moreover, since many features had missing values, we choose to drop columns with more than 30% missingness. After this procedure, we ended up with a dataset of 39 numerical features, such as systolic blood pressure and weight. Our target variable was readmission within 6 months.

D. UCI HEART DISEASE

This is the UCI[3] heart disease dataset available publicly online. This dataset origins in 1988 and consists of four databases: Cleveland, Hungary, Long Beach, and Switzerland. It originally contained 76 features, but only 14 are used for the task of classification. We use this 14-attribute subset, since the studies [36], [37] have also used 14 features. Five of these features are numerical and 9 categorical. The target variable is binary where a patient has either heart disease or not.

[2]https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset
[3]https://archive.ics.uci.edu/ml/datasets/heart+disease

VI. MISSING VALUE IMPUTATION

A. EXPERIMENTAL SETUP

To evaluate our methods, we first select a subset of each original dataset that has no missing values. We then introduce artificial missingness to these datasets and compare imputed values with the true ones. This step is performed using the library pyampute [38] which is a python adaptation of the R package ampute [39]. Inside, a multivariate amputation procedure is implemented, enabling the introduction of different missingness mechanisms separately or jointly. Using this package, we introduce missingness in proportions of 10%, 20%, 30%, 40% and 50% for all 3 missingness mechanisms. To evaluate these models, we employ 5-fold cross-validation and average the results between the 5 hold-out folds. The above procedure is repeated 10 times, and the results are averaged. In total, we train and evaluate each model $5 \times 10 = 50$ times. We use the Python library, namely scikit-learn [40], [41], for exploiting the simple approach (mentioned in Section IV-A), the k-NN (mentioned in Section IV-B), the MissForest (mentioned in Section IV-C), and the MICE method (mentioned in Section IV-D).

B. EVALUATION METRICS

We use normalized root mean squared error (NRMSE) for the numerical features. We choose this metric because we want to average the results of all continuous features for each experiment. If instead root mean squared error (RMSE) was used, attributes with larger values would overshadow smaller ones (e.g. Age is between 20-40 and cholesterol 125-300). We report our results using $mean \pm std$ of NRMSE.

C. RESULTS

In this section, we analyze the results for all four datasets regarding MAR, MCAR, and MNAR missingness mechanisms. We first discuss each dataset separately, and then we analyze the overall results and performance.

1) Results for the Framingham Heart Study Dataset

Results regarding this dataset are reported in Table 1. Firstly, for the MCAR case, we can see that the deep learning models outperform the standard baselines. The difference in mean NRMSE is $0.002 - 0.007$ where we observe a larger difference for higher percentages of missingness. We also see that the improved versions of NAA and GAIN are consistently better regarding the mean NRMSE and STD. Overall, the best performance is achieved by I-GAIN and I-NAA.

Regarding the MAR case, we see a larger difference between the proposed deep learning methods and the standard clinical baselines that can be up to 0.011 mean NRMSE. This is due to the fact that MAR is a more complicated missing pattern compared to MCAR and is harder for a model to incorporate and understand. Here we can see that I-NAA has consistently both the lowest mean NRMSE and STD.

Lastly, concerning the MNAR case, we also see that the deep learning methods perform more accurately for all missing rates. Moreover, I-NAA performs the best. Regarding the clinical baselines, MICE and MF perform better than kNN and Simple. We also observe that all methods perform worse as the missing rate increases, which is to be expected because a dataset with a higher missingness proportion is also harder to impute.

Overall, we also see that Simple and kNN produce the highest mean NRMSE, which is to be expected since these methods are relatively naive.

2) Results for the stroke dataset

The results for this dataset are reported in Table 2. The first thing we notice is that, in all cases, the deep learning approaches outperform the standard baselines. The difference regarding the MCAR and MAR cases is between 0.001 and 0.007 NRMSE. However, when it comes to the MNAR case the difference is much bigger and is up to ~ 0.2 for 50% missingness. It is also noted that, for the MAR case, Missforest and MICE perform almost as accurately as the deep learning approaches. This happens because, for this missing pattern, the missingness in numerical features is correlated with the categorical features, which are all binary. It is thus easier for a model to interpret these correlations compared to a case where missingness in numerical columns would be correlated to other numerical columns. The worst results are achieved by kNN and Simple, and in many cases, the margin is large. This happens because Simple imputes naively, and kNN is performing poorly for a dataset with mostly categorical variables. Concluding, we also see that the improvements and adjustments implemented on the deep learning imputation models reflect on the results too leading to more accurate imputations and less variability for almost all tests.

3) Results for the physionet heart disease dataset

Results for this dataset are reported in Table 3. The aforementioned dataset is more complex, having 39 continuous variables with high correlation. This is something that also reflects on the results. Here, we see that the differences between the standard clinical baselines and the proposed deep learning approaches are larger, which is to be expected since these neural networks are able to model complicated relationships more accurately. More specifically for the MCAR case, we see that the difference ranges between $\sim 0.012 - 0.02$ where bigger difference corresponds to larger missing rates. On the other hand, regarding the MAR and MNAR mechanisms, this difference is even greater, with a minimum of 0.015 for 10% missingness. The reason behind this phenomenon is that these patterns introduce additional complexity to the incomplete dataset and are thus harder for a model to understand. Also, since this dataset has only numerical features, the patterns introduced by MNAR and MAR are complicated compared to those of categorical features. Here, I-NAA and I-GAIN achieve the best results for all cases, proving the robustness of these approaches.

4) Results for the UCI disease dataset

Results for this dataset are reported in Table 4. Firstly, for all 3 missingness mechanisms and more specifically regarding lower missing rates we observe that the standard approaches of MICE and Missforest outperform the proposed methods by a small margin. It is noted, that this dataset is fairly small, namely ~ 1000 samples. This produces a challenge since approaches based on deep learning generally need large amounts of data during the training stage. Thus, this behavior is logical and expected. However, we see that as the missing rate increases and the imputation becomes harder, the deep learning models produce more accurate imputations. In more detail, we view a difference of up to 0.007 NRMSE for the MCAR case, 0.001 for the MAR case, and 0.022 for the MNAR case. We clearly see that for the last (MNAR) case the difference regarding the imputation performance is substantial which follows findings for the aforementioned datasets.

5) Overall remarks

Firstly, we see that in general, deep learning models outperform the standard baselines. The margin in performance is larger for the MAR and MNAR cases and a bit smaller for the MCAR pattern. This is due to the fact that the first two introduce additional complexity to the incomplete dataset, and the imputation task becomes harder for a method to model. From the experiments, we observe that this difference is greater when the relationships that define the features to be imputed (numerical features in our case) are more complex and also when the dataset has a higher number of samples. This is reasonable since deep learning models are

TABLE 1: Imputation results (Mean± STD of NRMSE) regarding the framingham heart study dataset for various missing rates. Best results per missing rate and missing data pattern are presented in bold.

Imputation method	10%	20%	30%	40%	50%
			MCAR		
Simple	0.112±0.0027	0.112±0.0031	0.113±0.0023	0.113±0.0020	0.130±0.0030
kNN	0.104±0.0033	0.107±0.0032	0.113±0.0018	0.117±0.0022	0.123±0.0036
MICE	0.099±0.0025	0.099±0.0035	0.111±0.0026	0.108±0.0031	0.111±0.0044
MF	0.101±0.0024	0.102±0.0031	0.105±0.0012	0.107±0.0030	0.113±0.0042
NAA	0.097±0.0052	0.099±0.0029	0.102±0.0016	0.103±0.0022	0.106±0.0033
I-NAA	0.097±0.0037	0.099±0.0018	**0.100±0.0020**	0.102±0.0020	**0.104±0.0020**
GAIN	0.097±0.0475	0.098±0.0431	0.103±0.0381	0.103±0.0391	0.107±0.0429
I-GAIN	**0.096±0.0438**	**0.097±0.0395**	0.103±0.0425	**0.101±0.0421**	0.105±0.0396
			MAR		
Simple	0.113±0.0024	0.113±0.0042	0.115±0.0052	0.113±0.0041	0.115±0.0027
kNN	0.110±0.0030	0.104±0.0032	0.110±0.0044	0.114±0.0047	0.120±0.0035
MICE	0.105±0.0029	0.103±0.0040	0.111±0.0049	0.110±0.0043	0.111±0.0029
MF	0.100±0.0026	0.103±0.0034	0.106±0.0051	0.109±0.0048	0.115±0.0033
NAA	0.101±0.0032	0.101±0.0027	0.103±0.0025	0.103±0.0024	0.105±0.0037
I-NAA	**0.099±0.0039**	**0.098±0.0023**	**0.099±0.0028**	**0.102±0.0036**	**0.104±0.0020**
GAIN	0.116±0.0456	0.100±0.0422	0.102±0.0415	0.103±0.0436	0.109±0.0493
I-GAIN	0.104±0.0420	0.098±0.0384	0.101±0.0383	0.102±0.0400	0.107±0.0460
			MNAR		
Simple	0.112±0.0028	0.112±0.0026	0.114±0.0031	0.113±0.0029	0.116±0.0046
kNN	0.100±0.0031	0.105±0.0024	0.114±0.0039	0.118±0.0030	0.124±0.0048
MICE	0.100±0.0031	0.103±0.0024	0.111±0.0040	0.109±0.0059	0.112±0.0051
MF	0.099±0.0030	0.102±0.0028	0.106±0.0035	0.106±0.0034	0.113±0.0063
NAA	0.097±0.0029	0.100±0.0035	0.101±0.0027	0.102±0.0024	0.105±0.0021
I-NAA	**0.097±0.0025**	**0.098±0.0017**	**0.101±0.0024**	**0.101±0.0011**	**0.104±0.0027**
GAIN	0.097±0.0447	0.102±0.0469	0.103±0.0414	0.104±0.0424	0.108±0.0465
I-GAIN	0.097± 0.0410	0.100±0.0428	0.103±0.0378	0.103±0.0391	0.106±0.0428

TABLE 2: Imputation results (Mean± STD of NRMSE) regarding the stroke dataset for various missing rates. Best results per missing rate and missing data pattern are presented in bold.

Imputation method	10%	20%	30%	40%	50%
			MCAR		
Simple	0.179±0.0114	0.190±0.0122	0.191±0.0084	0.195±0.0049	0.198±0.0050
kNN	0.167±0.0095	0.181±0.0079	0.190±0.0102	0.202±0.0065	0.206±0.0066
MICE	0.156±0.0073	0.166±0.0081	0.178±0.0109	0.197±0.0036	0.196±0.0064
MF	0.154±0.0087	0.168±0.0083	0.176±0.0099	0.185±0.0084	0.183±0.0062
NAA	0.153±0.0093	0.167±0.0080	**0.170±0.0098**	0.177±0.0065	0.180±0.0059
I-NAA	0.151±0.0101	**0.160±0.0090**	0.173±0.0098	**0.176±0.0077**	**0.176±0.0059**
GAIN	0.154±0.0503	0163±0.0625	0.172±0.0691	0.178±0.0672	0.181±0.0820
I-GAIN	**0.145±0.0465**	0.161±0.0561	0.172±0.0611	0.175±0.0631	0.180±0.0761
			MAR		
Simple	0.190±0.0112	0.193±0.0138	0.193±0.0139	0.194±0.0133	0.192±0.0122
kNN	0.172±0.0082	0.174±0.0110	0.179±0.0124	0.179±0.0118	0.181±0.0114
MICE	0.159±0.0077	0.162±0.0092	0.165±0.0118	0.167±0.0117	0.170±0.0132
MF	0.161±0.0071	0.161±0.0080	0.164±0.0103	0.163±0.0098	0.163±0.0093
NAA	0.154±0.0105	0.157±0.0151	0.160±0.0146	0.163±0.0137	0.163±0.0124
I-NAA	0.158±0.0120	0.153±0.0148	**0.158±0.0141**	**0.161±0.0128**	**0.162±0.0116**
GAIN	0.158±0.0640	0.159±0.0565	0.162±0.0647	0.163±0.0607	0.164±0.0727
I-GAIN	**0.153±0.0611**	**0.150±0.0505**	0.160±0.0570	0.162±0.0535	0.163±0.0650
			MNAR		
Simple	0.189±0.0110	0.187±0.0091	0.192±0.0045	0.197±0.0061	0.195±0.0073
kNN	0.172±0.0105	0.187±0.0082	0.192±0.0075	0.208±0.0079	0.213±0.0072
MICE	0.163±0.0057	0.179±0.0084	0.188±0.0047	0.199±0.0094	0.206±0.0102
MF	0.169±0.0090	0.178±0.0094	0.179±0.0063	0.189±0.0089	0.198±0.0079
NAA	0.162±0.0082	0.173±0.0097	0.175±0.0049	0.183±0.0046	0.184±0.0053
I-NAA	0.161±0.0101	0.170±0.0101	**0.170±0.0065**	**0.177±0.0051**	**0.176±0.0047**
GAIN	0.167±0.0529	0.174±0.0684	0.174±0.0777	0.188±0.0679	0.190±0.0725
I-GAIN	**0.157±0.0495**	**0.165±0.0609**	0.172±0.0708	0.184±0.0613	0.181±0.0631

TABLE 3: Imputation results (Mean± STD of NRMSE) regarding the physionet heart disease for various missing rates. Best results per missing rate and missing data pattern are presented in bold.

Imputation method	10%	20%	30%	40%	50%
			MCAR		
Simple	0.099±0.0063	0.101±0.0059	0.103±0.0056	0.105±0.0075	0.109±0.0082
kNN	0.090±0.0069	0.095±0.0065	0.098±0.0065	0.102±0.0115	0.106±0.0134
MICE	0.105±0.1854	0.114±0.1374	0.101±0.1374	0.102±0.0985	0.103±0.0881
MF	0.083±0.0066	0.086±0.0064	0.091±0.0064	0.094±0.0105	0.104±0.0118
NAA	0.074±0.0074	0.076±0.0060	0.079±0.0060	0.084±0.0100	0.086±0.0108
I-NAA	**0.071±0.0059**	**0.074±0.0063**	0.080±0.006	**0.081±0.0092**	**0.083±0.0097**
GAIN	0.078±0.0627	0.078±0.0602	0.079±0.0581	0.084±0.0591	0.091±0.0571
I-GAIN	0.071±0.0573	0.077±0.0565	**0.078±0.0547**	0.082±0.0541	0.089±0.0527
			MAR		
Simple	0.098±0.0063	0.098±0.0049	0.100±0.0050	0.102±0.0071	0.104±0.0085
kNN	0.090±0.0056	0.092±0.0048	0.095±0.0067	0.099±0.0099	0.103±0.0127
MICE	0.108±0.3132	0.116±0.2254	0.104±0.1851	0.101±0.1604	0.102±0.1437
MF	0.089±0.0067	0.094±0.0055	0.097±0.0068	0.101±0.0097	0.105±0.0125
NAA	0.073±0.0071	0.073±0.0055	0.076±0.0067	0.079±0.0082	0.082±0.0100
I-NAA	**0.071±0.0058**	**0.072±0.0045**	**0.074±0.0059**	**0.077±0.0068**	**0.080±0.0093**
GAIN	0.075±0.0551	0.078±0.0494	0.079±0.0457	0.080±0.0463	0.085±0.0495
I-GAIN	0.072±0.0507	0.073±0.0458	0.074±0.0414	0.079±0.0420	0.083±0.0455
			MNAR		
Simple	0.096±0.0070	0.099±0.0084	0.100±0.0084	0.102±0.0081	0.104±0.0097
kNN	0.085±0.0069	0.091±0.0087	0.094±0.0097	0.098±0.0108	0.103±0.0145
MICE	0.082±0.0483	0.083±0.0383	0.085±0.0400	0.088±0.0353	0.093±0.0330
MF	0.079±0.0065	0.083±0.0094	0.086±0.0108	0.089±0.0111	0.094±0.0142
NAA	0.070±0.0095	0.074±0.0092	0.077±0.0106	0.081±0.0118	0.084±0.0133
I-NAA	**0.068±0.0128**	0.072±0.0103	**0.074±0.0115**	**0.078±0.0116**	**0.081±0.0127**
GAIN	0.072±0.0569	0.074±0.0574	0.078±0.0560	0.082±0.0582	0.085±0.0627
I-GAIN	0.069±0.0522	**0.071±0.0523**	0.075±0.0501	0.079±0.0536	0.083±0.0581

TABLE 4: Imputation results (Mean± STD of NRMSE) regarding the UCI heart disease for various missing rates. Best results per missing rate and missing data pattern are presented in bold.

Imputation method	10%	20%	30%	40%	50%
			MCAR		
Simple	0.168±0.0195	0.168±0.0164	0.166±0.0141	0.167±0.0128	0.169±0.0124
kNN	0.153±0.0158	0.155±0.0130	0.158±0.0119	0.166±0.0175	0.165±0.0237
MICE	0.133±0.0189	0.135±0.0152	0.159±0.0239	0.163±0.0226	0.165±0.0215
MF	**0.129±0.0185**	**0.134±0.0152**	0.157±0.0133	0.160±0.0130	0.163±0.0139
NAA	0.132±0.0186	0.136±0.0152	0.138±0.0149	0.142±0.0141	0.145±0.0146
I-NAA	0.134±0.0191	0.135±0.0150	**0.136±0.0131**	**0.139±0.0131**	**0.142±0.0138**
GAIN	0.136±0.0768	0.138±0.0797	0.144±0.0741	0.148±0.0704	0.153±0.0693
I-GAIN	0.132±0.0704	0.136±0.0740	0.139±0.0694	0.142±0.0655	0.145±0.0633
			MAR		
Simple	0.162±0.0083	0.165±0.0072	0.167±0.0091	0.169±0.0106	0.169±0.0099
kNN	0.138±0.0074	0.139±0.0081	0.141±0.0097	0.142±0.0128	0.144±0.0122
MICE	0.142±0.0066	0.143±0.0063	0.147±0.0086	0.156±0.0183	0.161±0.0196
MF	**0.129±0.0076**	0.132±0.0083	0.134±0.0098	0.157±0.0120	0.159±0.0115
NAA	0.134±0.0052	0.134±0.0060	0.136±0.0067	0.138±0.0070	0.138±0.0066
I-NAA	0.134±0.0092	**0.132±0.0080**	**0.134±0.0080**	**0.135±0.0077**	**0.136±0.0071**
GAIN	0.136±0.0230	0.137±0.0202	0.140±0.0198	0.141±0.0151	0.142±0.0194
I-GAIN	0.135±0.0210	0.134±0.0182	0.139±0.0199	0.140±0.0126	0.141±0.0170
			MNAR		
Simple	0.162±0.0113	0.162±0.0113	0.165±0.0121	0.164±0.0113	0.166±0.0119
kNN	0.150±0.0151	0.150±0.0154	0.159±0.0141	0.167±0.0190	0.165±0.0241
MICE	**0.123±0.0077**	0.126±0.0137	0.157±0.0164	0.161±0.0161	0.163±0.0162
MF	0.127±0.0123	0.127±0.0123	0.156±0.0129	0.157±0.0124	0.161±0.0141
NAA	0.128±0.0102	0.128±0.0102	0.139±0.0115	0.141±0.012	0.145±0.0141
I-NAA	0.125±0.0102	**0.125±0.0102**	0.135±0.0112	**0.137±0.0111**	0.141±0.0138
GAIN	0.123±0.0252	0.129±0.0223	0.141±0.0239	0.147±0.0230	0.145±0.0273
I-GAIN	0.126±0.0214	0.126±0.0205	**0.135±0.0218**	0.138±0.0221	**0.140±0.0235**

known to need large datasets but are able to efficiently find patterns in data. However, regarding the GAIN approach, even though this model produces very accurate results, the STD is high. This occurs because it has two neural networks and thus a higher degree of variability, and it also has additional stochastic components such as the hint mechanism, which is generated randomly. All in all, the best performing model regarding almost all the experimental scenarios was I-NAA, which achieved the best NRMSE results. When it comes to the standard baselines, we see that MF and mice are competitive alternatives, achieving lower imputation results but not by a large margin. It is also noted that these methods require less data since the underlying mechanism is tree-based or a simple regression (e.g., linear). Lastly, Simple and kNN produce poor results with steadily poor NRMSE for all datasets and all missing rates. This happens because Simple treats each variable independently, ignoring feature relationships, and kNN is simply based on sample distance, which is not robust. With regard to the updated deep learning methods, namely I-GAIN and I-NAA, our main motivation was a comparison with the original versions for the imputation task. Results showed that I-GAIN outperforms GAIN and I-NAA outperforms NAA. For almost every dataset and for every missingness mechanism, we see that the proposed models achieve a lower mean of NRMSE compared to the original corresponding versions. This is more apparent with larger missing rates and for datasets that are more complex. Such cases are harder to impute, and a more robust model can handle them more accurately. It is also worth noting that in almost all cases, the proposed models achieve lower STD compared to the primary models, which again proves their effectiveness.

VII. POST-IMPUTATION PREDICTION

A. EXPERIMENTAL SETUP

To further evaluate our models, we also conduct post-imputation prediction, where we test each one of the aforementioned methods. These methods are first trained using the MCAR pattern and subsequently used to impute the pre-existing missing values from the datasets. For every model, we save the complete, imputed dataset and perform a simple random forest to predict the target variable using 5-fold cross validation and Synthetic Minority Over-sampling Technique [42] (SMOTE) for each fold's training set (since the produced complete datasets are highly unbalanced).

B. EVALUATION METRICS

For the post-imputation prediction task we use the F1-score since target variables in all four datasets are imbalanced.

C. RESULTS

Results for the post-imputation prediction are reported in Table 5. For each dataset produced by the imputation methods, we demonstrate F1-score results for the prediction task.

Firstly, for the UCI heart disease dataset, we observe that the imputation results are high, which is foreseeable since the variable of interest is easy to predict. The methods that performed better for the imputation perform better for this task too. Here, I-GAIN produces the best results, achieving a difference of $\sim 9\%$ compared to the clinical baselines. The improved deep learning versions also achieve better results compared to their original versions. kNN and Simple have the worst results which is anticipated since the corresponded datasets have been imputed naively.

For the physionet dataset we see that the best performing model is I-NAA with an F1-score of $\sim 48\%$. Regarding all clinical baselines, we notice a lower score with a minimum difference of $\sim 2.5\%$ compared to I-NAA. We also see that I-NAA and I-GAIN perform better than NAA and GAIN for this step, validating that the introduced adjustments are better. Moreover, the F1-score attained by every method is relatively low, which is due to the imbalance of the target variable.

Results concerning the stroke dataset can be viewed on the third row of Table 5. Here, the best-performing method is I-GAIN, achieving $\sim 2\%$ more f1-score compared to MICE. It is also apparent that the improved deep learning models produce better scores compared to their original versions. Also, kNN is the worst-performing method that follows the imputation results since this is a poor method for a dataset with many categorical variables. In general, the F1-score is low, which is to be expected since this dataset is hard to predict due to reasons such as class imbalance, low correlation between input and output variables etc.

Framingham is the last dataset we used for this step. The best outcome is produced by I-NAA with an F1-score $\sim 2\%$ higher compared to the most commonly used methods within the clinical framework. The remarks made on the previous datasets are apparent in Framingham too. More specifically, we see that the improved deep learning approaches achieve a better f1-score, while simple kNN yields the worst results. Overall, we observe that I-NAA and I-GAIN produce the best results for the post-imputation task, which is something that is observed for the imputation task too. We also see that for all EHR datasets, the improved versions of NAA and GAIN are more robust. Also, the naive approaches achieve the lowest results, which is logical since their imputations were previously shown to be suboptimal. It can also be seen that random forest produces low predictive results for physionet, stroke and framingham. This happens because these datasets are highly imbalanced, and SMOTE is not enough to remedy this. For this scenario, more complex pre-processing steps and models should be used, but for the purpose of this study, a comparison is feasible without these.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we studied the case of missing value imputation for EHR datasets. We built upon existing deep learning architectures by introducing improvements regarding the training procedure and the architecture. We evaluated them for both imputation and post-imputation performance. Regarding the

TABLE 5: Post imputation accuracy (F1-score) for all four datasets. Best result per dataset is presented in bold.

Dataset \ Model	Simple	kNN	MICE	MF	NAA	I-NAA	GAIN	I-GAIN
UCI	88.26	89.00	90.89	89.99	96.33	98.29	96.76	**99.33**
Physionet	44.89	45.67	44.21	45.53	46.32	**47.99**	46.72	47.55
Stroke	31.93	31.32	32.18	31.78	32.58	33.51	33.12	**33.92**
Framingham	44.32	45.19	45.52	45.38	47.73	**48.33**	45.25	47.85

task of missing value imputation, we experimented with various missing rates and various missing patterns, showing that the introduced approaches outperformed the state-of-the-art ones, reaching better normalized RMSE compared to the clinical baselines. For the prediction task (post-imputation prediction), findings suggested that our introduced approaches achieved the best imputation F1-scores for all four datasets with a difference of up to $\sim 9\%$ compared to other approaches.

In the future, we aim to evaluate our methods with more medical datasets as well as test more deep learning approaches such as variational autoencoders and attention-based models. Moreover, since the proposed methods are trained and tested on the same dataset, they have been essentially fitted to a specific patient distribution. For this reason, a model trained on one of these four datasets could be used for missing value imputation on a different EHR dataset to assess cross-dataset performance.

REFERENCES

[1] Kristiina Häyrinen, Kaija Saranto, and Pirkko Nykänen. Definition, structure, content, use and impacts of electronic health records: A review of the research literature. International journal of medical informatics, 77:291–304, 06 2008.

[2] Martin Cowie, Juuso Blomster, Lesley Curtis, Sylvie Duclaux, Ian Ford, Fleur Fritz, Samantha Goldman, Salim Janmohamed, Jörg Kreuzer, Mark Leenay, Alexander Michel, Seleen Ong, Jill Pell, Mary Southworth, Wendy Stough, Martin Thoenes, Faiez Zannad, and Andrew Zalewski. Electronic health records to facilitate clinical research. Clinical research in cardiology : official journal of the German Cardiac Society, 106, 01 2017.

[3] Edward Kennedy, Wyndy Wiitala, Rodney Hayward, and Jeremy Sussman. Improved cardiovascular risk prediction using nonparametric regression and electronic health record data. Medical care, 51, 12 2012.

[4] Yilong Zhang, Zachary Zimmer, Lei Xu, Raymond L. H. Lam, Susan Huyck, and Gregory Golm. Missing data imputation with baseline information in longitudinal clinical trials. Statistics in Biopharmaceutical Research, 14(2):242–248, 2022.

[5] Brian J Wells, Kevin M Chagin, Amy S Nowacki, and Michael W Kattan. Strategies for handling missing data in electronic health record derived data. EGEMS (Washington, DC), 1(3):1035, 2013.

[6] Rubin DB. Multiple imputation for nonresponse in survey. John Wiley & Sons, 2004.

[7] SWJ Nijman, AM Leeuwenberg, I Beekers, I Verkouter, JJL Jacobs, ML Bots, FW Asselbergs, KGM Moons, and TPA Debray. Missing data is poorly handled and reported in prediction model studies using machine learning: a literature review. Journal of Clinical Epidemiology, 142:218–229, 2022.

[8] Tan Duy Le, Razvan Beuran, and Yasuo Tan. Comparison of the most influential missing data imputation algorithms for healthcare. In 2018 10th International Conference on Knowledge and Systems Engineering (KSE), pages 247–251, 2018.

[9] Olawale Ayilara, Lisa Zhang, Tolu Sajobi, Richard Sawatzky, Eric Bohm, and Lisa Lix. Impact of missing data on bias and precision when estimating change in patient-reported outcomes from a clinical registry. Health and Quality of Life Outcomes, 17, 06 2019.

[10] Daniel J. Stekhoven and Peter Bühlmann. MissForest—non-parametric missing value imputation for mixed-type data. Bioinformatics, 28(1):112–118, 10 2011.

[11] Melissa Azur, Elizabeth Stuart, Constantine Frangakis, and Philip Leaf. Multiple imputation by chained equations: What is it and how does it work? International journal of methods in psychiatric research, 20:40–9, 03 2011.

[12] Lovedeep Gondara and Ke Wang. Mida: Multiple imputation using denoising autoencoders. In Dinh Phung, Vincent S. Tseng, Geoffrey I. Webb, Bao Ho, Mohadeseh Ganji, and Lida Rashidi, editors, Advances in Knowledge Discovery and Data Mining, pages 260–272, Cham, 2018. Springer International Publishing.

[13] Emily Kogan, Kathryn Twyman, Jesse Heap, Dejan Milentijevic, Jennifer Lin, and Mark Alberts. Assessing stroke severity using electronic health record data: A machine learning approach. BMC Medical Informatics and Decision Making, 20, 01 2020.

[14] Helena Aidos and Pedro Tomás. Neighborhood-aware autoencoder for missing value imputation. In 2020 28th European Signal Processing Conference (EUSIPCO), pages 1542–1546, 2021.

[15] Konstantinos Psychogyios, Loukas Ilias, and Dimitris Askounis. Comparison of missing data imputation methods using the framingham heart study dataset. In 2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), pages 1–5, 2022.

[16] M.N. Ramli, Ahmad Shukri Yahaya, Nor Ramli, Noor Faizah Fitri Md Yusof, and Mohd Mustafa Al Bakri Abdullah. Roles of imputation methods for filling the missing values: A review. Advances in Environmental Biology, 7:3861–3869, 10 2013.

[17] Senthilkumar Mohan, Chandrasegar Thirumalai, and Gautam Srivastava. Effective heart disease prediction using hybrid machine learning techniques. IEEE Access, 7:81542–81554, 2019.

[18] Saba Bashir, Zain Sikander Khan, Farhan Hassan Khan, Aitzaz Anjum, and Khurram Bashir. Improving heart disease prediction using feature selection approaches. In 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), pages 619–623, 2019.

[19] Safial Ayon, Md Islam, and Rahat Hossain. Coronary artery heart disease prediction: A comparative study of computational intelligence techniques. IETE Journal of Research, 68:2488–2507, 01 2020.

[20] Chitra Jegan. Heart attack prediction system using fuzzy c means classifier. IOSR Journal of Computer Engineering, 14:23–31, 01 2013.

[21] Mehak Gupta, Thao-Ly T. Phan, H. Timothy Bunnell, and Rahmatollah Beheshti. Obesity prediction with ehr data: A deep learning approach with interpretable elements. ACM Trans. Comput. Healthcare, 3(3), apr 2022.

[22] Kelvin Kwakye and Emmanuel Dadzie. Machine learning-based classification algorithms for the prediction of coronary heart diseases, 2021.

[23] Rajni Bhalla and Amandeep Bagga. Rb-bayes algorithm for the prediction of diabetic in "pima indian dataset", 12 2019.

[24] Jingshu Liu, Zachariah Zhang, and Narges Razavian. Deep ehr: Chronic disease prediction using medical notes. In Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, Proceedings of the 3rd Machine Learning for Healthcare Conference, volume 85 of Proceedings of Machine Learning Research, pages 440–464. PMLR, 17–18 Aug 2018.

[25] Aixia Guo, Randi E. Foraker, Robert M. MacGregor, Faraz M. Masood, Brian P. Cupps, and Michael K. Pasque. The use of synthetic electronic health record data and deep learning to improve timing of high-risk heart failure surgical intervention by predicting proximity to catastrophic decompensation. Frontiers in Digital Health, 2, 2020.

[26] Suraj Kumar Gupta, Aditya Shrivastava, SP Upadhyay, and Pawan Kumar Chaurasia. A machine learning approach for heart attack prediction. Intelligent Sustainable Systems), pages 741–747, 2022.

[27] José M. Jerez, Ignacio Molina, Pedro J. García-Laencina, Emilio Alba, Nuria Ribelles, Miguel Martín, and Leonardo Franco. Missing data imputation using statistical and machine learning methods in a real breast cancer problem. Artificial Intelligence in Medicine, 50(2):105–115, 2010.

[28] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: Missing data imputation using generative adversarial nets. In Jennifer Dy and Andreas Krause, editors, Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pages 5689–5698. PMLR, 10–15 Jul 2018.

[29] Weinan Dong, Daniel Fong, Jin-sun Yoon, Eric Wan, Laura Bedford, Eric Tang, and Cindy Lam. Generative adversarial networks for imputing missing data for big data clinical research. BMC Medical Research Methodology, 21, 04 2021.

[30] Sungkyu Park, Cheng-Te Li, Sungwon Han, Cheng Hsu, Sang Won Lee, and Meeyoung Cha. Learning sleep quality from daily logs. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, page 2421–2429, New York, NY, USA, 2019. Association for Computing Machinery.

[31] Boseong Seo, Jaekyung Shin, Taejin Kim, and Byeng D. Youn. Missing data imputation using an iterative denoising autoencoder (idae) for dissolved gas analysis. Electric Power Systems Research, 212:108642, 2022.

[32] Daniel J. Stekhoven and Peter Bühlmann. MissForest—non-parametric missing value imputation for mixed-type data. Bioinformatics, 28(1):112–118, 10 2011.

[33] Stef van Buuren. Multiple imputation of discrete and continuous data by fully conditional specification. Statistical Methods in Medical Research, 16(3):219–242, 2007. PMID: 17621469.

[34] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, page 5769–5779, Red Hook, NY, USA, 2017. Curran Associates Inc.

[35] Zhongheng Zhang, Linghong Cao, Rangui Chen, Yan Zhao, Lukai Lv, Ziyin Xu, and Ping Xu. Electronic healthcare records and external outcome data for hospitalized patients with heart failure. Scientific Data, 8:46, 02 2021.

[36] Ankur Gupta, Rahul Kumar, Harkirat Singh Arora, and Balasubramanian Raman. Mifh: A machine intelligence framework for heart disease diagnosis. IEEE Access, 8:14659–14674, 2020.

[37] Ghulab Nabi Ahmad, Hira Fatima, Shafi Ullah, Abdelaziz Salah Saidi, and Imdadullah. Efficient medical diagnosis of human heart diseases using machine learning techniques with and without gridsearchcv. IEEE Access, 10:80151–80173, 2022.

[38] Rianne M Schouten, Davina Zamanzadeh, and Prabhant Singh. pyampute: a python library for data amputation, August 2022.

[39] Rianne Margaretha Schouten, Peter Lugtig, and Gerko Vink. Generating missing values for simulation purposes: a multivariate amputation procedure. Journal of Statistical Computation and Simulation, 88(15):2909–2930, 2018.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[41] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pages 108–122, 2013.

[42] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. J. Artif. Int. Res., 16(1):321–357, jun 2002.

KONSTANTINOS PSYCHOGYIOS is currently pursuing his M.Eng. of Electrical and computer engineering from National Technical University of Athens (NTUA), Greece.

He is also concurrently working in the industry as a machine learning engineer implementing and researching solutions such as IDS for federated systems and GAN for image generation. His research interests lie in the areas of cybersecurity and bioinformatics using machine learning techniques. He has also published and presented research papers in international conferences organized by reputable organizations (e.g. IEEE).

LOUKAS ILIAS received the integrated master's degree (five-year studies) from the School of Electrical and Computer Engineering at the National Technical University of Athens (NTUA). He has completed a research internship at University College London (UCL).

He is currently pursuing the Ph.D. degree with the Decision Support Systems Laboratory of the School of Electrical and Computer Engineering at NTUA. He is currently a Researcher with the Decision Support Systems Laboratory, NTUA, where he is involved in EU-funded research projects. His research interests include computational linguistics, computer vision, speech processing, natural language processing, social media analysis, and detection of complex brain disorders, including Alzheimer's disease, epilepsy, sleep disorders, etc. He has published in numerous journals, including IEEE Journal of Biomedical and Health Informatics, Expert Systems With Applications (Elsevier), Applied Soft Computing (Elsevier), Computer Speech & Language (Elsevier), and Frontiers in Aging Neuroscience, and has presented his research at international conferences, including IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI'22).

CHRISTOS NTANOS is a research director and senior researcher, and holds a Bachelor in Electronic and Computer Engineering, a Master in Electronic and Computer Engineering obtained at the University of Birmingham, UK, and a Master's in Business Administration (MBA) from the Athens University of Economics and Business. He holds a PhD from the School of Electrical and Computer Engineering at the National Technical University of Athens (NTUA).

Dr. Ntanos is very experienced in managing ICT and cross-cutting research and implementation projects, including the coordination of ones under the Horizon 2020 programme. In this capacity, he is coordinating "Search and Rescue: Emerging technologies for the Early location of Entrapped victims under Collapsed Structures & Advanced Wearables for risk assessment and First Responders Safety in SAR operations". He is the coordinator of the MES-CoBraD project to build a multidisciplinary expert system for the assessment and management of Complex Brain Disorders and the coordinator and scientific manager of "Sphinx - A Universal Cyber Security Toolkit for Health-Care Industry", which aims at assessing and reducing cyber risks in hospitals and care centres to protect privacy/data/infrastructure. He has also coordinated ChildRescue, which involves NGOs from Greece and Belgium to enhance the processes of locating and tracking missing children and children in migration. Dr. Ntanos has significant expertise in the management, design, and development of ICT solutions, analogous and parametric estimating of IT and database systems, context awareness, risk management, Decision Support Systems, Business Process Reengineering, Knowledge Representation, eGovernment and eParticipation. He has a sixteen-year experience contributing in various research and implementation projects from the managerial, as well as the design and implementation perspectives.

DIMITRIS ASKOUNIS is a Professor at the School of Electrical and Computer Engineering of the National Technical University of Athens (NTUA) and Deputy Director of the Decision Support Systems Laboratory.

He has over 25 years of experience in the fields of decision support systems, intelligent information systems and manufacturing, e-business, e-government, open and linked data, big data analytics and AI algorithms as well as the application of modern IT techniques in the management of companies and organizations.

He has been the scientific director of over 50 European research projects in the above areas (FP7, Horizon2020, etc). He has participated in many projects and activities in NIS and MEDA countries related to the monitoring and evaluation of major projects, training of business executives, development of IT systems, etc.

For a number of years he was the advisor to the Minister of Justice and the Special Secretary for Digital Convergence for the introduction of information and communication technologies in public administration. Since June 2019 he is President of the Information Society SA.