A Comparative Study of Missing Data Imputation Methods for Activity Recognition Task

Afia Sajeeda*, B M Mainul Hossain and Sumon Ahmed

Institute of Information Technology, University of Dhaka, Dhaka-1000

**E-mail: bsse0832@iit.du.ac.bd*

Received on 20 March 2021, Accepted for publication on 13 October 2021

ABSTRACT

While data is integral to address real-world problems using machine learning techniques, the availability of complete data is challenging as data goes missing during collection or even afterward. Missing values make data unsuitable for use and demand imputation techniques to resolve the problem. Here, we compare the performance of four existing missing data imputation techniques KNN, MICE, GAIN, and HexaGAN by applying them on KU-HAR dataset and two datasets of the collection, Nurse Care Activity Recognition Dataset. Our investigation suggests that HexaGAN learns the original data distribution better and demands further considerations, as the RMSE between the imputed data and the real data is lower. To investigate the role of activation function, we replace the underlying ReLU activation functions of the neural networks in HexaGAN architecture with the Swish activation function. Experimental results show that the modified version of HexaGAN possesses the potential to outperform the original one when applied to the same activity recognition datasets.

Keywords: Generative Adversarial Networks, GANs, Missing Data, Imputation, Deep Learning.

1. Introduction

While meticulous approaches are used during the collection of real-world data, gathering real-world data without any missing values remains a challenging task. Diligently collecting data does not guarantee the absence of missing values within data [1]. The missing values within data may or may not exhibit distinctive patterns. These patterns can be used to map relations between the observable and non-observable variables. According to [2], based on the exhibition of patterns or simply, the distribution of missing values, missing data can be classified into three categories: Missing Completely at Random (MCAR), Missing at Random (MAR), Missing Not At Random (MNAR). For MCAR data, the missing values do not show any particular behavior or pattern. No conclusion can be drawn or variable-to-variable mapping can be extracted in such circumstances. When data is missing at random, the observable variables resemble some pattern. When data is MNAR, both observable and non-observable variables bear particular characteristics. Regardless of the category of missing data, its presence makes a considerably huge volume of data to become unusable for machine learning-oriented studies and degrades the performance of machine-learning models. Imputation techniques are needed to prevent vast portions of data from becoming unusable for machine-learning-oriented tasks.

While imputation techniques using Generative Adversarial Networks (GANs) [3] have been recently gaining attention [4, 5, 6], conventional approaches are still widely recognized as the state-of-the-art [7, 8, 9, 10]. Conventional or adversarial, whichever category the imputation techniques fall under, face a major challenge when the amount of available complete data is considerably low. Conversely saying, when the missing rate of data is as high as 80%-90%, imputation performance begins to degrade. The purpose of this study is to investigate if a modification in the architectures of imputation models can contribute to better imputation performance. More specifically, we observe the impact of the change of activation functions on HexaGAN, an imputation model based on Generative Adversarial Networks. For the imputation task, two activity recognition datasets from the CARE- COM Nurse Care Dataset [26] collection is selected. The activity recognition task itself is deemed challenging because it requires actions performed by an agent (nurse) on another agent (patient) to be characterized based on the movement of the first agent (nurse). A third Human Activity Recognition (HAR) dataset is extracted from a subset of KU-HAR [30] which originally contains data belonging to 18 categories of simple human activities like standing, sitting, running, walking, and lying.

Our contribution can be stated as follows:

- Modification of an existing approach and analysis of the imputation performance.
- Comparison among four existing approaches with the newly proposed approach.
- Comparison of all five approaches with different rates of missing data.
- List of possible impacts of activation functions (ReLU versus Swish) on imputation tasks.
- Building an imputation model for a challenging activity recognition dataset in the MCAR missing data scenario.

In our work, using four different existing imputation models, we performed imputation on two nurse care activity recognition datasets and one simple human activity datasets, with different missing rates and computed the RMSE to assess the deviation between the original data and the imputed values. We also prepared a test dataset containing complete and imputed data and calculated the accuracy using pre-trained (on complete data) classifiers (KNN, Decision Tree, and Random Forest) to investigate if the imputed data aligns with the correct target class. Motivated

by the performance of HexaGAN [4], our experiment was extended to modifying the architecture of HexaGAN by substituting the ReLU activation function within the neural network architecture of HexaGAN, with the Swish activation function and computing the root mean square error (RMSE) and accuracy for the modified setting. Experimental results show that the modified setting of HexaGAN outperforms the original, 6 out of 10 times.

The remainder of this paper is organized as follows: Section 2 provides related work, followed by the Methodology in Section 3 and Experiment Setup in Section 4. Results are discussed in Section 5. Lastly, Section 6 drops a concluding note and future direction.

2. Related Works

Before Generative Adversarial Networks (GANs) demonstrated game-changing performance in imputation tasks, conventional approaches competed with each other to be crowned state-of-the art of missing data imputation tasks. K-Nearest Neighbors [7], Matrix Completion [8], Multivariate Imputation by Chained Equations (MICE) [9], Denoising Autoencoders [10] fall in the second category.

K Nearest Neighbor, commonly known as KNN, is a simple approach where at first Euclidean distance is computed and compared with the nearest k neighbors. KNN was implemented as an imputation technique for DNA microarrays in [7]. Experimental results in [7] have shown KNN to be robust to the percentage of missing data with a deviation of 6-26% from real values and was thus recommended as a standard approach.

MICE takes a multi-step approach. [9] shows the demonstration of version 2.9 of MICE on a simple dataset containing four variables. At first, several copies of the dataset are made and in each copy, the missing values are imputed temporarily based on the available remaining data. The mean for each of the independent variables in the dataset are computed and used for temporarily imputing for each of the corresponding independent variables. Next, taking other independent variables as features, values for another independent variable are predicted. This process is repeated until the dataset has statistical estimates (predictions) for all independent variables. To put it simply, MICE focuses on imputing one variable at a time using features of the other variables present in the data.

GAN-based approaches constitute modifications of the vanilla GAN architecture. The most basic GAN architecture constitutes a Generator Network and a Discriminator Network. The generator model outputs fake samples from random noise to replicate the original data distribution. The Discriminator Model attempts to aid the generator by monitoring the samples produced by the generator. The Discriminator does this by identifying the fake samples from the real ones and dispatching feedback to the generator in the form of a loss function value. Using this loss value, a generator improves its performance. Some noteworthy variations of GANs include WGAN [12], StyleGAN [13], CycleGAN [14], LSGAN [15] and CGAN [16]. Initially intended to generate images, the applications of GANs have extended to interesting use- cases such as face-manipulation, image-to-image translation, text-to-image translation, video prediction [17] and missing data imputation.

GAIN [6] is the allegedly first Generative Adversarial Network-oriented approach to the missing data problem. Following GAIN, other GAN-oriented solutions have been proposed from time to time, for example, MISGAN [5] and HexaGAN [4]. The target of the generator in GAIN is to generate a vector of imputed values taking the real data vector with missing values, corresponding mask (indicator of missing values), and noise as input to the generator. The discriminator is given the task to identify which components are real and which components are imputed. The GAIN framework approached the missing data problem in five real-world datasets from the UCI machine learning repository (Breast, Spam, Letter, Credit, and News) [18] by predicting the mask vector.

HexaGAN [4] framework, overall, addresses three real-world problems simultaneously: missing data, class imbalance, and missing labels. Of the six components, three are involved in the missing data problem: the encoder and one set of generator-discriminator. In addition to MNIST handwritten dataset, HexaGAN was used to impute data from the UCI machine learning repository similar to GAIN [6]. The working mechanism of HexaGAN would be futher discussed in the following section.

GAIN-GTEx is another follow-up of GAIN with some additional leverages for gene expression imputation [19]. Stackelberg GAN takes note of mode collapse and dropping issues of GAN by using multiple generators instead of the standard single generator [20]. High missing rates of data are considered in GAMIN [21] which also presents a confidence prediction mechanism. CollaGAN [22] handles the complicated nature of images by converting the image imputation problem to a multi-domain image-to-image translation task. This enables the missing data to be successfully estimated using the remaining clean data set. An approach to impute temporal information or multi-stage information is given by end-to-end generative model E^2 GAN [23]. Luo et. al used a modified Gated Recurrent Unit [24] with GAN for multivariate time series imputation. SciGAN [25] framework has been proposed for scRNA sequence imputations by alleviating the previous problems of oversmoothening and removal of cell-to-cell stochasticity. Other works on imputation based on GAN include [26] and [27]. For a comprehensive study on the use of GANs for missing data imputation, the reader can refer to [28].

3. Methodology

In our work, we take the initiative to draw a comparison between four existing imputation techniques, namely, KNN, MICE, GAIN, and HexaGAN primarily and then, being motivated by the performance of HexaGAN, we modify the architecture of the underlying neural network that might result in a boost to the imputation performance. HexaGAN [4] is based on Generative Adversarial Networks (GANs). GANs are deep learning-based techniques that learn the underlying data distribution and thus can be used in data imputation tasks. For the purpose of imputation, we borrow three elements from HexaGAN, which are: the Encoder, the Generator, and the Discriminator, all of which are neural networks.

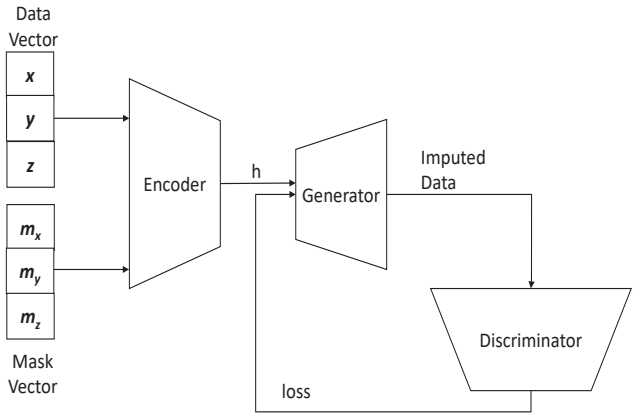

Fig. 1. Overview of models working together

Table 1: Activation functions in the underlying layers of HexaGAN and HexaGAN-Swish (Modified Architecture)

	Encoder	**Generator**	**Discriminator**
HexaGAN	Hidden=ReLU Output=ReLU	Hidden=ReLU Output=Sigmoid	Hidden=ReLU Output=None
Hexa-GAN-Swish	Hidden=**Swish** Output=**Swish**	Hidden=**Swish** Output=Sigmoid	Hidden=**Swish** Output=none

Activation functions equip neural networks with the ability to handle non-linear data. It determines which neuron will fire information and which neuron will hold back during forward propagation. ReLU, Sigmoid function, and Tanh functions are the most widely used activation functions, especially ReLU. ReLU is more like the de-facto standard. In 2017, Google Brain has brought a new activation function into the arena- the *Swish* Activation Function [29]. Swish is different from ReLU in terms of handling extremely negative values. In the case of ReLU, all negative values are zeroed. Meanwhile, Swish zeroes values that are extremely negative. To put it simply, Swish accommodates a more few negative input values as non-zero, in comparison to ReLU. There is a possibility that these small negative values carry information significant to imputation tasks- which we investigate in our approach.

In Table 1, we show the modifications in terms of activation functions of neural network layers from the original neural network architecture of HexaGAN using bold fonts.

Training Process: Suppose **X** is a complete data vector, **m** is a mask vector indicating the position of missing elements and **z** is a noise vector. The dimensions of all the vectors are equal. By element-wise multiplication of **X** and **m**, an incomplete data vector i.e. data vector with missing values is obtained. The encoder takes **X**, **m** and **z** as input and computes $\tilde{\mathbf{x}}$. Equation 1 shows the computation of $\tilde{\mathbf{x}}$ similar to [4].

$$\tilde{\mathbf{x}} = \mathbf{m} \odot \mathbf{x} + (1 - \mathbf{m}) \odot \mathbf{z} \quad (1)$$

The encoder *E* uses $\tilde{\mathbf{x}}$ as shown in Figure 1 in generating a hidden variable **h**. The generator takes **X**, $\tilde{\mathbf{x}}$,**m** and **h** as input to give imputed data vector $\hat{\mathbf{x}}$. Equation 2 shows the computation of $\hat{\mathbf{x}}$ similar to [4]. Now the discriminator tries to distinguish between imputed data vector $\hat{\mathbf{x}}$ and complete data vector **X**. Training continues until training loss is converged.

$$\hat{\mathbf{x}} \leftarrow \mathbf{m} \odot \mathbf{x} + (1 - \mathbf{m}) \odot \bar{\mathbf{x}} \quad (2)$$

Algorithm 1: Missing Data Imputation
Input **x** - data with missing values
m - vector indicating missing elements
z - noise vector sampled from $U(0,1)$
Output $\hat{\mathbf{x}}$ - imputed data

repeat
Sample a batch of pairs (**x, m, z**)
$\tilde{\mathbf{x}} = \mathbf{m} \odot \mathbf{x} + (1 - \mathbf{m}) \odot \mathbf{z}$
$\mathbf{h} \leftarrow E(\tilde{\mathbf{x}}, \mathbf{m})$
$\bar{\mathbf{x}} \leftarrow G(\mathbf{h})$
$\tilde{\mathbf{x}} \leftarrow \mathbf{m} \odot \mathbf{x} + (1\text{-}\mathbf{m}) \odot \mathbf{z}$
$\hat{\mathbf{x}} \leftarrow \mathbf{m} \odot \mathbf{x} + (1\text{-}\mathbf{m}) \odot \bar{\mathbf{x}}$
Update Discriminator *D* using Stochastic Gradient Descent (SGD)
$\nabla D \mathcal{L} D + \lambda_1 \mathcal{L}$ Gradient Penalty GP
Update Encoder *E* and Generator *G* using SGD
$\nabla E \mathcal{L} G + \alpha_1 \mathcal{L}_{recon}$
$\nabla G \mathcal{L} G + \alpha_1 \mathcal{L}_{recon}$
until Training loss is converged

Algorithm 1 (originally proposed in [b4]) describes missing data imputation and the corresponding model training. The two lines of pseudocode shown within the box indicate the

point where we tweak the network in order to investigate difference of performance from the original.

4. Experiment Setup: Dataset Description and Pre-processing

Nurse Care Activity Recognition Dataset: The collection consists of 3 datasets. The datasets are records on six nursing care activities which is available at (https://ieee-dataport.org/competitions/nurse-care- activity-recognition-challenge). Information of these six activities along with their corresponding labels and frequencies are given in Tables 2 and 3. Three sensor de- vices were used to record these activities: motion capture sensor, meditag sensor, and accelerometer sensor. The data collected by three sensors are organized in separate files as 3 datasets [11]. For our study, we experimented with the data collected by the accelerometer sensor and the meditag sensor.

Accelerometer Dataset: The accelerometer sensor which was positioned in the right chest pocket of the nurse collects the x, y, and z coordinates of the respective point. Details of the dataset are noted down in Table 2.

Table 2: Classes of data with corresponding frequencies for Nurse Care-Accelerometer Dataset

Activity Description	Activity	Frequency
Vital Signs Measurement	2	7382
Blood Collection	3	9839
Blood Glucose Measurement	4	6142
Drip Retention & Connection	6	3744
Oral Care	9	5410
Diaper Exchange & Cleaning	12	7466
Total=	-	39983

Meditag Dataset : The meditag sensor notes down the two dimensional positions, x and y of a bluetooth device which the nurse carries in his/her chest pocket and the air pressure in mHg. Details of the dataset are noted down in Table 3.

Table 3: Classes of data with corresponding frequencies for Nurse Care-Meditag Dataset

Activity Description	Activity	Frequency
Vital Signs Measurement	2	2393
Blood Collection	3	2843
Blood Glucose Measurement	4	2009
Drip Retention & Connection	6	1571
Oral Care	9	1619
Diaper Exchange & Cleaning	12	4717
Total=	-	15152

Table 4: Classes of data with corresponding frequencies for KU-HAR Dataset

Activity Description	Activity	Frequency
Stand	0	10000
Sit	1	10000
Lay	5	10000
Walk	11	10000
Run	14	10000
Total=	-	50000

KU-HAR Dataset: This Human Activity Recognition (HAR) dataset [30] holds data on 18 different activities collected using smartphone sensors (Accelerometer and Gyroscope). For the experiment, out of 18 classes we used 10,000 data from each of the 5 classes, namely, stand, sit, lay, walk, and run. A total of 8 features are used to describe an activity. These are- durations of time for recording data by the Accelerometer and Gyroscope, acceleration along X, Y, Z axes and rate of rotation around the X, Y, Z axes. The dataset is available at (https://data.mendeley.com/datasets/45f952y38r/3).

Pre-processing and Settings: Data of different percentages of missing values, 20%, 50%, and 80% were used in our experiments. The 5 models were trained separately with each of the different missing rates and imputation performance was assessed.

5. Results and Discussion

To investigate the performance of four missing data imputation techniques, i.e., KNN, MICE, GAIN, and HexaGAN, they have been applied on three datasets described in the previous section with varying missing rates. The performance evaluation is primarily carried out by measuring the root mean square error (RMSE) between the actual data values and the imputed values. Experimental results show a reduction in RMSE scores for GAN-based approaches such as GAIN, HexaGAN, and HexaGAN-Swish; indicative of the imputed values being closer to the actual data values. Next, 5-fold cross-evaluation is conducted by computing the accuracy of pre-trained classifiers (KNN, decision tree, and random forest) on test data containing both real data and imputed data values. The accuracy of the pre-trained classifiers on test data, which has been imputed by GANs gives higher scores implying GANs implicitly learn the properties of original data better. The knowledge of these properties, in turn, helps to align data to its corresponding class.

Root Mean Square Error: Root Mean Square Error (RMSE) is used to investigate the imputation performance based on the difference between the original complete data and the imputed data. 5-fold cross-validation is carried out to mitigate overfitting. It is evident from Tables 5, 6 and 7 that RMSE values consistently show a decreasing trend for GAN-based approaches across missing rates of 20%, 50%

and 80%. Lower RMSE values suggest that HexaGAN learns the real data distribution better by at least 46.72% even when the percentage of missing values in the dataset can go as high as 80% (Table 5). The possible reason is the underlying encoder. Encoders enable the overall architecture to learn the data representations better.

Table 5: RMSE and Accuracy obtained by using the five approaches on Nurse Care-Accelerometer data with different missing rates

Metric	Missing Rate	KNN	MICE	GAIN	HexaGAN	HexaGAN-Swish
Mean Test RMSE	20%	1.142	0.1784	0.1452	0.0337	0.0355
	50%	1.8893	2.0419	0.1741	0.0761	0.0701
	80%	2.5598	2.7479	0.1971	0.1050	0.0981
Mean Test Accuracy by KNN	20%	0.1833	0.1802	0.2935	0.2754	0.269
	50%	0.1852	0.183	0.2624	0.2602	0.2602
	80%	0.1846	0.1889	0.2478	0.1803	0.1557
Mean Test Accuracy by Decision Tree	20%	0.1761	0.1741	0.4948	0.3363	0.3307
	50%	0.1771	0.1785	0.4178	0.2511	0.2497
	80%	0.1847	0.1717	0.382	0.2393	0.2262
Mean Test Accuracy by Random Forest	20%	0.1796	0.1787	0.4948	0.3363	0.3432
	50%	0.1843	0.1863	0.4332	0.2582	0.2572
	80%	0.1995	0.1943	0.3985	0.218	0.1672

Table 6: RMSE and Accuracy obtained by using the five approaches on Nurse Care-Meditag data with different missing rates

Metric	Missing Rate	KNN	MICE	GAIN	HexaGAN	Hexa-GAN-Swish
Mean Test RMSE	20%	0.6248	0.6685	0.3172	0.0853	0.0837
	50%	1.0244	1.1307	0.2787	0.1530	0.1563
Mean Test Accuracy by KNN	20%	0.3337	0.337	0.6649	0.4325	0.501
	50%	0.3382	0.3365	0.5618	0.3386	0.5279
Mean Test Accuracy by Decision Tree	20%	0.3674	0.3638	0.7735	0.5247	0.5484
	50%	0.3743	0.3684	0.6494	0.4345	0.5271
Mean Test Accuracy by Random Forest	20%	0.3668	0.3678	0.7855	0.5742	0.5401
	50%	0.372	0.3678	0.6517	0.4743	0.5692

Table 7: RMSE and Accuracy obtained by using the five approaches on KU-HAR data on 20% missing data

Metric	Missing Rate	KNN	MICE	GAIN	HexaGAN	HexaGAN-Swish
Mean Test RMSE	20%	3.7544	2.0776	0.0927	0.0422	0.0392
Mean Test Accuracy by KNN	20%	4e-05	5e-05	0.7624	0.4621	0.4729
Mean Test Accuracy by Decision Tree	20%	0.00011	0.0002	0.8234	0.5571	0.5738
Mean Test Accuracy by Random Forest	20%	0.00014	0.0001	0.861	0.581	0.6219

We also tweak the architecture of HexaGAN replacing ReLU activation layers with Swish activation layers as demonstrated in Table 1. In the introductory paper of Swish [27], the authors have pointed out reasons why Swish may outperform ReLU. Swish is bounded below and unbounded above. This constrains extremely negative values to be zeroed whereas, in ReLU, all negative values are zeroed. ReLU, thus might not be able to capture significant patterns which Swish might be able to, in case the values are not extremely negative. Experimental results mentioned in Table 5, 6 and 7 demonstrates that Swish performs better than ReLU when applied to HexaGAN as the mean test RMSE for HexaGAN-Swish is lower. In terms of RMSE, for every 10 times, the modified architecture outperforms the original 6 times.

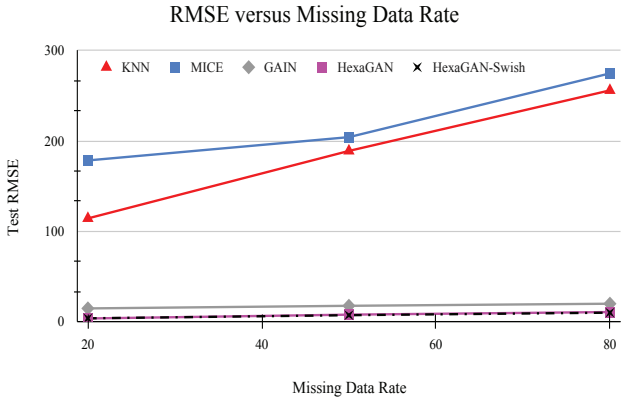

Fig. 2. Test RMSE for Nurse Care-Accelerometer Dataset against different rates of missing data for all five approaches

The graph in Figure 2 shows that the RMSE for all five approaches increases with the increase of the missing rate for Nurse Care-Accelerometer dataset. Despite the high rate of missing values, the slope of the line for GAN-based approach is much less than conventional methods, i.e., KNN and MICE. The line for GAIN is parallel with the line for HexaGAN and HexaGAN-Swish. HexaGAN and HexaGAN-Swish almost coincide along every point.

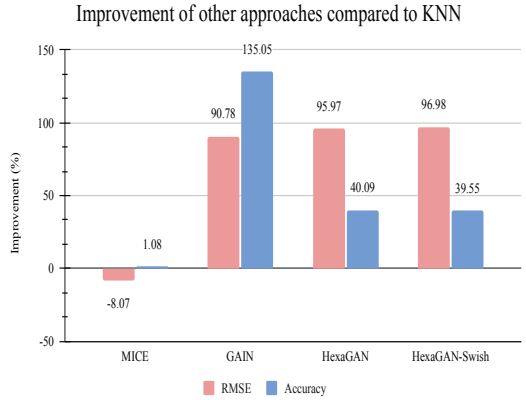

Fig. 3. Comparison of RMSE and Accuracy for Nurse Care-Accelerometer Dataset using Random Forest Classifier when 50% missing data is imputed

We have also analyzed the comparative performance improvement for each method. KNN is a state-of-the-art method for imputation. We compare the performance of MICE, GAIN, HexaGAN, and HexaGAN-Swish with KNN (number of k-neighbors is considered as 10) in terms of RMSE of the imputed data from the real complete data. From Figure 3, we see MICE has seen a degradation of 8.07% rather than improvement. Lowering the number of k-neighbors in KNN, enables MICE to outperform KNN. The GAN-based approaches (GAIN, HexaGAN, and HexaGAN-Swish) have shown above 90% improvement.

Accuracy: To further assess the imputation performance, we align the imputed data with their corresponding labels (activity numbers) and used three state-of-the-art pre-trained classifiers which are KNN, Decision Tree, and Random Forest to check whether these classifiers can align these data to the corresponding activity classes. The classifiers are pre-trained on completed labeled data i.e., data with no missing values. The test data contains both original data and imputed data. To minimize overfitting, 5-fold cross-validation is performed for the classification task as well.

From Tables 5, 6 and 7, it is evident that GAIN consistently gives the highest accuracy value. HexaGAN provides a slightly better score than data generated by HexaGAN-Swish for the accelerometer dataset. However for meditag (Table 6) and KU-HAR (Table 7) data, HexaGAN-Swish produces a higher value of accuracy. In Figure 3, we show the improvement in accuracy considering the accuracy on KNN as baseline.

We would like to report that the classification accuracy found using GAN-imputed data is better than the accuracy obtained using real and complete data (0% missing rate). For instance, from Table 5, we find the accuracy ranging from 26%-43% for GAN-imputed data for different classifiers when missing rate is 50%. Meanwhile, the accuracy using real data are 17.73%, 17.5% and 17.88 % for KNN, Decision Tree and Random Forest classifiers respectively. This implies that GAN-based approaches have the ability to predict the imputed data and are comparatively less affected by the noise inherent in real data.

6. Conclusion

Missing data imputation techniques are proposed to eradicate the missing data problem. With the goal of finding an improvement among notable existing approaches, the imputation task was performed on two datasets from CARE-COM Nurse Care Activity Recognition Datasets. An additional obstacle introduced by the two datasets is that nurse care activity concerns an action performed by one person on another person. To be straightforward, the imputation model faces the challenge of imputing values that will abide by the complex feature-label relationship. The third dataset, KU-HAR has simpler feature-label relationships in comparison being a data on individual activity with no external agent dependency.

We carried out the imputation task using KNN, MICE, GAIN, and HexaGAN at first and reported the test RMSE. Like its precedent attempts [4], HexaGAN outperformed the others. Intending to investigate ways of doing better, we tweaked the architecture of HexaGAN by replacing the existing ReLU activation functions with the Swish activation function. The motivation behind doing so is that we wanted to assess whether non-extreme negatives values that are converted to zero by ReLU activation function in the original HexaGAN framework bear any meaning or feature which could improve the imputation performance. Swish, unlike ReLU, handles negative data by forcing extreme negative values to zero but accommodating non-extreme negative values.

Conducting our experiment on different missing rates, we see instances where the modified architecture of HexaGAN with the Swish activation function outperforms the original. In terms of RMSE, the modified network performs better than the original 6 out of 10 times. This evokes the possibility that the modification may open scopes for improvement and recommends further investigation to gather more evidence.

Furthermore, to examine whether GAN-imputed data respects the expected behavior of their original labels (activity numbers), we used three state-of-the-art pre-trained classifiers which are KNN, Decision Tree and Random Forest to calculate the accuracy of imputation. Our results show that the accuracy ranges from 26%-43% for GAN-imputed data and a little beyond 17% for real data for different classifiers. The phenomenon summons the possibility of GAN-centric approaches being less affected by noise prevalent in real data as well as class imbalance problem which is prevalent in Nurse Care-Accuracy and Nurse Care- Meditag datasets. While we used the basic Random Forest classifier to report accuracy, Random Forest-based Learning-To-Rank Algorithms (LtR) studied in previous works [31, 32] may be considered for assessment of performance of imputed data when training data is subject to class imbalance.

Acknowledgement

This research has been supported by the ICT Division, Government of the People's Republic of Bangladesh through the ICT Innovation Fund for the year 2020-21.

References

1 D. B. Rubin, "Inference and Missing Data," *Biometrika*, vol. 63, no. 3, p. 581, 1976, Available: https://doi.org/10.2307/2335739.

2 S. van Buuren, *Flexible Imputation of Missing Data, Second Edition*. Second edition, CRC Press, 2019, Available: https://doi.org/10.1201/9780429492259.

3 I. Goodfellow *et al.*, "Generative Adversarial Nets," 2014, Available: https://doi.org/10.1145/3422622.

4 U. Hwang, D. Jung, and S. Yoon, "HexaGAN: Generative Adversarial Nets for Real World Classification," 2019.

5 S. C.-X. Li, B. Jiang, and B. M. Marlin, "MISGAN: Learning from Incomplete Data with Generative Adversarial Networks," 2019.

6 J. Yoon, J. Jordon, and M. van der Schaar, "Gain: Missing Data Imputation Using Generative Adversarial Nets," in *Proceedings of the 35th International Conference of Machine Learning*, pp. 5689–5698, 2018

7 O. Troyanskaya *et al.*, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001, Available: https://doi.org/10.1093/bioinformatics/17.6.520.

8 T. Hastie, R. Mazumder, J. D. Lee, and R. Zadeh, "Matrix Completion and Low-rank SVD via Fast Alternating Least Squares," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 3367–3402, 2015.

9 S. van Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate Imputation by Chained Equations in R," *Journal of Statistical Software*, vol. 45, no. 3, 2011, Available: https://doi.org/10.18637/jss.v045.i03.

10 P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and Composing Robust Features with Denoising Autoencoders," in *ICML '08: Proceedings of the 25th International Conference on Machine Learning*, pp. 096-1103, 2008, [Online; accessed 16-February-2021]. Available: https://doi.org/10.1145/1390156.1390294

11 Sozo Inoue, Paula Lago, Shingo Takeda, Alia Shamma, Farina Faiz, Nattaya Mairittha, Tittaya Mairittha, "Nurse Care Activity Recognition Challenge", IEEE Dataport, 2019, Available: https://dx.doi.org/10.21227/2cvj-bs21.s

12 M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," in *International Conference on Machine Learning*, pp. 214–223, 2017.

13 T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks." in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4401-4410, 2019.

14 J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks." in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223-2232, 2017.

15 X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley "Least squares generative adversarial networks." in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2794-2802.

16 M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," 2014. [Online; accessed 16-February-2021]. Available: https://doi.org/10.48550/arXiv.1411.1784

17 J. Brownlee, *Generative Adversarial Networks with Python*. Machine Learning Mastery, 2019.

18 D. Dua and C. Graff, 2019, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

19 C. Li, K. Xu, J. Zhu, J. Liu and B. Zhang, "Triple Generative Adversarial Networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 9629-9640, 2022, Available: https://doi.org/10.1109/TPAMI.2021.3127558.

20 H. Zhang, "Medical Missing Data Imputation by Stackelberg GAN," 2018. [Online; accessed 16-February-2021]. Available: https://www.ml.cmu.edu/research/dap-papers/f18/dap-zhang-hongyang.pdf

21 S. Yoon and S. Sull, "GAMIN: Generative Adversarial Multiple Imputation Network for Highly Missing Data." in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8456-8464, 2020.

22 D. Lee, J. Kim, W. Moon, and J. C. Ye, "CollaGAN: Collaborative GAN for missing image data imputation." in P*roceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2487-2496, 2019.

23 Y. Luo, Y. Zhang, X. Cai, and X. Yuan,"E2GAN: End-to-End Generative Adversarial Network for Multivariate Time Series Imputation." in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, AAAI Press, pp. 3094-3100, 2019.

24 Y. Luo, X Cai, Y. Zhang, and J. Xu, "Multivariate time series imputation with generative adversarial networks." In *Advances in Neural Information Processing Systems*, pp. 1596- 1607, 2018

25 Y. Xu, Z. Zhang, L. You, J. Liu, Z. Fan, and X. Zhou, "scIGANs: single-cell RNA-seq imputation using generative adversarial networks," *Nucleic Acids Research*, Jun. 2020, Available: https://doi.org/10.1093/nar/gkaa506

26 A. Kazemi and H. Meidani, "IGANI: Iterative Generative Adversarial Networks for Imputation With Application to Traffic Data," *IEEE Access*, vol. 9, pp. 112966–112977, 2021,Available: https://doi.org/10.1109/access.2021.3103456.

27 R. Viñas, T. Azevedo, E. R. Gamazon, and P. Liò, "Deep Learning Enables Fast and Accurate Imputation of Gene Expression," *Frontiers in Genetics*, vol. 12, Apr. 2021, Available: https://doi.org/10.3389/fgene.2021.624128

28 J. Kim, D. Tae, and J. Seok, "A survey of missing data imputation using generative adversarial networks", In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, IEEE, pp. 454-456, 2020.

29 P. Ramachandran, B. Zoph, and Q. Le, "Swish: A Self-Gated Activation Function," 2017.

30 A. Nahid, N. Sikder and I. Rafi, "KU- HAR: An Open Dataset for Human Activity Recognition", 2020

31 Mendeley Data, V3, Available: https://doi.org/10.17632/45f952y38r.3 [Online; accessed 16-February-2021].

32 M. Ibrahim "Sampling non-relevant documents of training sets for learning-to-rank algorithms.", *International Journal of Machine Learning and Computing*, vol. 10, no. 2, 2020.

33 M. Ibrahim," Reducing correlation of random forest based learning-to-rank algorithms using subsample size", *Computational Intelligence*, vol. 35, no. 4, pp. 774-798, 2019.