Denoising Probabilistic Diffusion Models for Synthetic Healthcare Image Generation

1st Antonella Iuliano
Department of Mathematics, Computer Science and Economics
University of Basilicata
Potenza, Italy
antonella.iuliano@unibas.it

2nd Pietro Liò
Department of Computer Science and Technology
University of Cambridge
Cambridge, UK
pl219@cam.ac.uk

3rd Gilda Manfredi
Department of Mathematics, Computer Science and Economics
University of Basilicata
Potenza, Italy
gilda.manfredi@unibas.it

4th Federico Romaniello
Department of Mathematics, Computer Science and Economics
University of Basilicata
Potenza, Italy
federico.romaniello@unibas.it

2024 IEEE International Workshop on Metrology for Living Environment (MetroLivEnv) | 979-8-3503-8501-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/METROLIVENV60384.2024.10615511

***Abstract*—Healthcare data are an essential resource in Machine Learning (ML) and Artificial Intelligence (AI) to improve clinical practice, empower patients and enhance drug development with the aim to discover new medical knowledge. In particular, the biomedical imaging analysis plays a important role in the healthcare context producing a huge amount of data that can be used to study complex diseases and their evolution in a deeper way or to predict their onsets. In this work we consider an approach based on Denoising Diffusion Probabilistic Models (DDPM) which is a type of generative model that uses a parameterized Markov chain and variational inference to generate synthetic samples that match real data. In particular, we execute a study by training on Malaria images and generating high-quality synthetic samples in order (i) to test the performance of the DDPMs, (ii) to estimate the association between original and synthetic data and (iii) to understand how the natural and human-made environmental factors impact Malaria disease. Finally, we use a well-defined convolutional neural network for classification tasks to assess the DDPM's goodness in generating the synthetic images.**

***Index Terms*—Denoising Probabilistic Diffusion Models, Healthcare dataset, Deep Learning, Synthetic data**

I. Introduction

Machine Learning (ML) and Artificial Intelligence (AI) algorithms are continuously evolving to solve complex problems and enhance our understanding of data. A special interest is devoted to the Diffusion Models (DMs), which have gained significant attention for their ability to capture and simulate composite processes like data generation and image synthesis (see, for instance, [21]). In these fields diffusion refers to a specific approach for generating data using a stochastic process similar to a Markov chain. In particular, DMs are used to create new data based on that they are trained on and then gradually transforming it into more complicated and realistic data. This transformation is achieved through a sequence of invertible operations.

Diffusion Probabilistic Models (DPMs) have attracted considerable attention in the field of medical imaging due to their well-established mathematical explications, adversarial-free training strategy, and ability to achieve stable and controlled generation. Diffusion-based generative models were first introduced in [17], and then popularized by [5], through the introduction of the "Denoising Diffusion Probabilistic Models (DDPMs)". Specifically, DDPMs are a class of generative models that work by iteratively adding noise to an input signal (image, text, or audio) and then learning to denoise from the noisy signal to generate new samples.

The main field of application of DDPMs is medical imaging where there is a strong need of tools to support and improve the clinical routines for patients. They arose from 2021 to 2023 with particular attention to the Magnetic Resonance Imaging (MRI) as a powerful tool for diagnosis and therapy (see, for instance, [3]). In fact, the complexity of data collection process, the lack of experts, the patient privacy, and the difficulty to require the authorization create a problem in the annotation procedure of medical imaging. Therefore, DPMs are very useful since they permit to generate synthetic medical images to mitigate data security and to alleviate the question of medical data scarcity concerning public health datasets. For instance, in [8] the authors show that DPMs can integrate high-quality medical data for MRI and computed tomography (CT) improving the performance of ML models and its quality.

Another interesting application is related to the generative models for Structure-based Drug Design (SBDD). See, for example, [4], where the authors present a tool for flexible small-molecule design and optimization based on DiffSBDD and [16], where the authors formulate a 3D-conditioned generation problem, which aims to generate different ligands with high binding affinity for specific protein targets. Moreover, in [7] is introduced a generative model for proteins that directly sample novel protein structures and sequences. Relevant is also the approach illustrated in [19], where the authors develop a novel motif-scaffolding procedure based on DPM for the design of

vaccines and enzymes.

Motivated by these and others new recent works based on DDPMs, the purpose of this work is to examine the impact of the natural and human-made environmental factors on Malaria disease by using data presented in [14]. Specifically, the novelty of this study is developing a DDPM-based approach to generate synthetic medical images. To assess the goodness of the generated data, we use a well-known Convolutional Neural Network (CNN), trained to be able to classify the data. Here, we use as study case a dataset characterized by parasite-infected and uninfected red blood cells images. Therefore, our approach allowing us to have a better understanding of how synthetic information can be applied to (i) improve ML and AI models, (ii) complete existing datasets when there is no or little data and (iii) increase their accessibility without compromising the privacy of individual patients in the dataset (see, for instance, [2], and [20]).

Overall, this study permit to assess that the synthetic images are very similar to real images and can be used in training models for medical research. Additionally, integrating both synthetic and real data helps us to improve the performance in a variety of medical applications, as synthetic data is a powerful enhancement of realistic information.

II. Methods

DDPMs was introduced in [5] are a type of generative model that uses a parameterized Markov chain and variational inference to generate synthetic samples that match real data. In particular, they are latent variable models of the form

$$p_\theta(\boldsymbol{x}_0) := \int p_\theta(\boldsymbol{x}_{0:T})\mathrm{d}\boldsymbol{x}_{1:T}, \tag{1}$$

where $x_1, \ldots, x_T$ are latents of the same dimensionality as the data $x_0 \sim q(x_0)$. The joint distribution $p_\theta(\boldsymbol{x}_{0:T})$ is called the *reverse process*, and it is defined as a Markov chain with learned Gaussian transitions starting at $p(\boldsymbol{x}_T) = \mathcal{N}\big(\boldsymbol{x}_T; \boldsymbol{0}, \boldsymbol{I}\big)$ (i.e., an isotropic gaussian distribution that does not depend on θ):

$$p_\theta(\boldsymbol{x}_{0:T}) := p(\boldsymbol{x}_T)\prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t), \tag{2}$$

with

$$p_\theta(\boldsymbol{x}_{t-1}|x_t) := \mathcal{N}\big(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)\big),$$

where $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$ are two functions parametrised by θ to be learned. The approximate posterior $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$, called the *forward process* or *diffusion process*, is a Markov chain that gradually adds Gaussian noise to the data according to a variance schedule $\beta_1, \ldots, \beta_T$, i.e.,

$$q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) := \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \tag{3}$$

with

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) := \mathcal{N}\big(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\boldsymbol{I}\big).$$

In other words, β_t indicates that at each step the trade-off between information to be kept from the previous step and new noise to be added. The training phase is performed by optimizing the usual variational bound on negative log likelihood as follows:

$$\begin{aligned} L :&= \mathbb{E}\big[-\log p_\theta(\boldsymbol{x}_0)\big] \leq \mathbb{E}_q\bigg[-\log\frac{p_\theta(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\bigg] \\ &= \mathbb{E}_q\bigg[-\log p(\boldsymbol{x}_T) - \sum_{t\geq 1}\log\frac{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\bigg]. \end{aligned} \tag{4}$$

The *forward process* variances β_t is learned by reparameterization described in [10] or held constant as hyperparameters, and expressiveness of the *reverse process* is ensured in part by the choice of Gaussian conditionals in $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$. A remarkable property of the *forward process* is that it admits sampling $\boldsymbol{x}_t$ at an arbitrary time t in a closed form. Specifically, we have

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\big(\boldsymbol{x}_t; \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0, (1-\overline{\alpha}_t)\boldsymbol{I}\big). \tag{5}$$

where $\alpha_t := 1-\beta_t$ and $\overline{\alpha}_t := \prod_{s=1}^{t}\alpha_s$. Denote $D_{KL}(q||p)$ as the Kullback–Leibler (KL) divergence from distribution p to distribution q. The learning of the *reverse process* consists in finding $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$ that minimise the KL divergence between $q(\boldsymbol{x}_0)$ and $p_\theta(\boldsymbol{x}_0)$, or equivalently that minimise the negative log-likelihood of $p_\theta(\boldsymbol{x}_0)$ under $q(\boldsymbol{x}_0)$. Mathematically, we want to find $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$ that minimise:

$$\boldsymbol{\mu}_\theta^*, \boldsymbol{\Sigma}_\theta^* = \mathrm{argmin}_{\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta}\big(D_{KL}\big(q\big(\boldsymbol{x}_0\big)||p(\boldsymbol{x}_0)\big)\big). \tag{6}$$

Specifically, reworking the expression in (6) in terms of L, we minimise the upper bound of L using a stochastic gradient descent (see, for more details, [5]) and rewriting (4) as:

$$\begin{aligned} &\mathbb{E}\bigg[\underbrace{D_{KL}\big(q(\boldsymbol{x}_T|\boldsymbol{x}_0)||p\big(\boldsymbol{x}_T\big)\big)}_{L_T} \\ &+\sum_{t>1}\underbrace{D_{KL}q\big(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\big)}_{L_{t-1}}\underbrace{-\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)}_{L_0}\bigg] \end{aligned} \tag{7}$$

where L_T doesn't depend on$\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$ and so doesn't need to be considered in the optimisation process; L_0 is not hard to optimise. Whereas, the KL divergence L_{t-1} is used to compare $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ against forward process posterior when the following conditions hold:

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\big(\boldsymbol{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\beta}_t\boldsymbol{I}\big), \tag{8}$$

with

$$\begin{aligned} &\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) := \frac{\sqrt{\overline{\alpha}_{t-1}}\beta_t}{1-\overline{\alpha}_t}\boldsymbol{x}_0 + \frac{\sqrt{\alpha}_t(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t}\boldsymbol{x}_t, \\ &\overline{\beta}_t := \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t. \end{aligned} \tag{9}$$

Hence, since all KL divergences in (7) are comparisons between Gaussians, they can be calculated in a Rao-Blackwellized fashion with closed form expressions instead of high variance Monte Carlo estimates. Now, in order to further simplify the task of the model in practice, we set the variance

of the reverse process in (2) to an untrained time dependent constant such that

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_\theta(\boldsymbol{x}_t, t), \sigma_t^2 \boldsymbol{I}). \tag{10}$$

Reminding (8) , the term L_{t-1} in (7) is rewrite using a specific parameterization

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2{\sigma_t}^2} ||\tilde{\mu_t}(\boldsymbol{x}_t, \boldsymbol{x}_0) - \mu_\theta(\boldsymbol{x}_t, t)||^2 \right] + C \tag{11}$$

where C is a constant that does not depend on θ. We can then see that each KL divergence term of the upper bound corresponds to a given time step and its optimisation simply consist in minimising the L_2 distance between the model and the mean of the reverse process conditioned on x_0, both evaluated at the considered time step. Hence, we can expand (11) further by reparameterizing (5) as $\boldsymbol{x}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\varepsilon_t$, for $\varepsilon_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and applying the *forward process posterior* formula (9):

$$\begin{aligned} &L_{t-1} - C \\ &= \mathbb{E}_{\boldsymbol{x}_0, \varepsilon_t} \left[\frac{1}{2{\sigma_t}^2} \left\| \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}} \varepsilon_t \right) - \mu_\theta(\boldsymbol{x}_t, t) \right\|^2 \right]. \end{aligned} \tag{12}$$

From (12) we have that μ_θ must predict $\frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}} \varepsilon_t \right)$ given $\boldsymbol{x}_t$ choosing as parameterization

$$\mu_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}} \varepsilon_0(\boldsymbol{x}_t, t) \right), \tag{13}$$

where ε_0 is a function approximator intended to predict ε from $\boldsymbol{x}_t$. Therefore, we have To sample $\boldsymbol{x}_{t-1} \sim p_\theta(\boldsymbol{x}_{t-1} \| \boldsymbol{x}_t)$ is to compute $\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}} \varepsilon_0(\boldsymbol{x}_t, t) \right) + \sigma_t \boldsymbol{z}$, where $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Furthermore, with the parameterization (13), (12) simplifies to:

$$\begin{aligned} &D_{KL}\big(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) || p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\big) \\ &= \mathbb{E}_{\boldsymbol{x}_0, \varepsilon_t} \left[\frac{\beta_t^2}{2{\sigma_t}^2 \alpha_t (1-\overline{\alpha}_t)} ||\varepsilon_t - \varepsilon_0(\boldsymbol{x}_t, t)||^2 \right], \end{aligned} \tag{14}$$

which resembles denoising score matching over multiple noise scales indexed by t (see, for instance, [18]). The minimisation of each KL divergence terms of the upper bound given in (14) now consists in minimizing the L_2 distance between the model and the noise.

III. Applications

To generate synthetic images that represent both infected and uninfected cells, we trained the DDPM model with Malaria dataset [14], which is a major global health threat. The dataset contains a total of 27.558 cell images with equal instances of parasitized and uninfected cells. This dataset is taken from the official National Library of Medicine website [11]. The process for generating the synthetic images is summarised in Figure 1, while some samples of the generated images are shown in Figure 2. We trained the DDPM model for 100 epochs on a NVIDIA Geforce RTX 4090 GPU. We used a batch size of 32 and the Adam optimization algorithm [9]. We implemented two different algorithms, one for training the model and the other one for inference the trained network to generate the synthetic images [5].

To test the goodness of the generated images, we adopted a popular image classification, image segmentation, and object detection module, named You Only Look Once version 8 (YOLOv8)[15]. This model became popular for its high speed and accuracy while maintaining a small model size. In particular, we use the same Malaria dataset to train a YOLOv8 classification model that has been pre-trained with the ImageNet dataset [6]. The technique of using a pre-trained model on a similar problem as a starting point is commonly used to quickly optimize models and achieve high accuracy even with a relatively small dataset [12]. The YOLOv8 classification model has been trained for 100 epochs with a batch size of 16. We used Stochastic Gradient Descent as the optimization algorithm, with a learning rate of 0.01 and a momentum of 0.937, which were indicated in [15] as the best hyperparameters. In Figure 3 we show samples of how the YOLOV8 architecture classifies the cell images. The confusion matrix in Figure 4 is determined for the performance evaluation of the YOLOv8 used after the classification. In particular, we compute the accuracy of a model using the given formula

$$ACC = \frac{TN + TP}{TN + FP + FN + TP}$$

where: TP indicates the True Positives, namely the cases that have been predicted as positive and they indeed have that disease; TN indicates the True Negatives, namely the cases that have been predicted as negative and they indeed do not have that disease; FP indicates the False Positives, namely the cases that have been predicted as positive but they do not have that disease; FN indicates the False Negatives, namely the cases that have been predicted as negative but they have that disease.

The value of the accuracy is equal to 0.97

The YOLOv8 performances are illustrated in Figure 5 where we show the training and validation loss curves. These outcomes indicate an optimal fit, i.e., a model that does not overfit or underfit. After evaluating the performance of YOLOv8 with the Malaria dataset, we decided to use a set of images generated by DDPM as input for YOLOv8 to test the effectiveness of DDPM in generating synthetic images for augmenting small datasets. To have a quantitative evaluation metric of the DDPM's goodness, we input to the YOLOv8 network a set of 1000 synthetic images and a set of 1000 unseen images taken from Malaria dataset. Furthermore, we apply the Chi-square and Kullback-Leibler (KL) divergence tests to compare the distributions of the Malaria and synthetic data. The p-values of Chi-square and KL divergence tests shown in Figure 6 indicate a good-quality of synthetic images.

From preliminary observations, we can observe that increasing the number of diffusion steps enhances the DDPM's ability to learn denoising, although at the cost of longer training times. On the other hand, reducing the number of diffusion steps leads to behavior indicative of overfitting. Specifically,

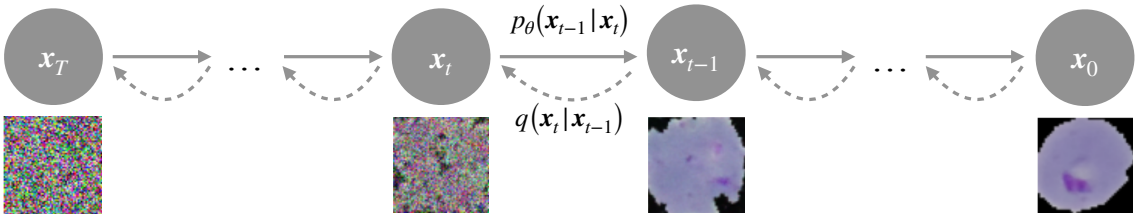

Fig. 1. The DDPMs procedure to generate Malaria synthetic data.

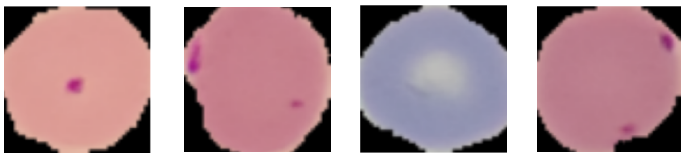

Fig. 2. Synthetic Malaria images generated by DDPMs trained for 100 epochs on a NVIDIA Geforce RTX 4090 GPU.

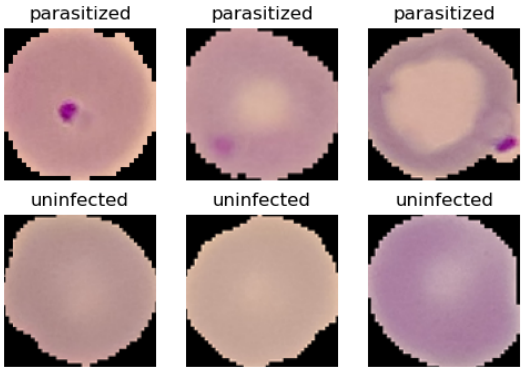

Fig. 3. Samples of Malaria cells obtained from a subset of the Malaria dataset that was not utilized during the training phase. The cells were classified using YOLOv8.

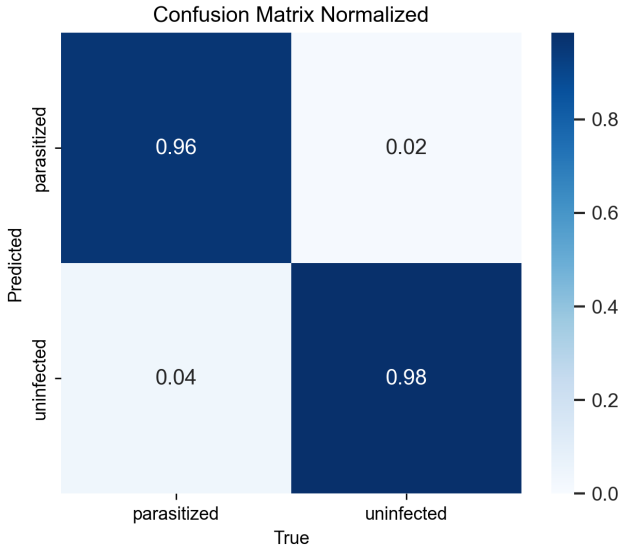

Fig. 4. Confusion matrix for binary classification.

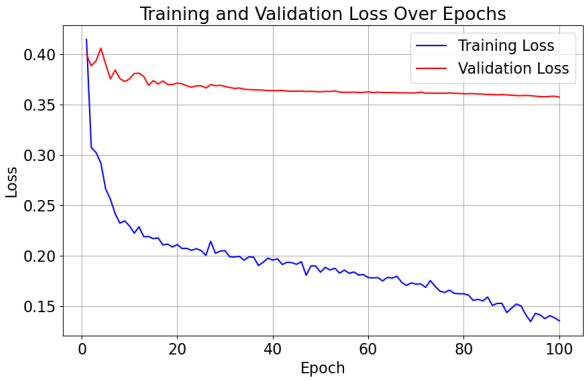

Fig. 5. YOLOv8 training and validation losses.

the DDPM generates synthetic images that are significantly different from the training dataset. To validate our outcomes, the next steps will be to explore SHAP method (SHapley Additive exPlanations) for image classification to assess the quality of synthetic Malaria images from DDPMs. Therefore, this analysis will provide the basis for future developments and for the understanding of natural and human-made environmental factors that impact Malaria disease as an effective diagnostic aid.

A. *Implementation*

For the implementation and training of both the CNN and the DDPM architectures, we utilized Pytorch [13] and TensorFlow [1]: two end-to-end open-source platforms for machine learning that offer Python APIs.

IV. Conclusions

The preliminary results of this work aim to contribute to a more specific understanding of how DDPMs can be used to generate synthetic healthcare data bridging the gap between algorithmic design and application. In particular, we first produce the synthetic data of a well-known Malaria dataset using DDPM, and then we validate the goodness of the results by applying a YOLOv8 classification deep learning model. Some quantitative metrics are also discussed. Therefore, the final aim will be to provide a valuable starting point for those who want to exploit DDPMs in real-world scenarios that require the generation of images or videos and how synthetic data can enhance the accuracy of classification AI models.

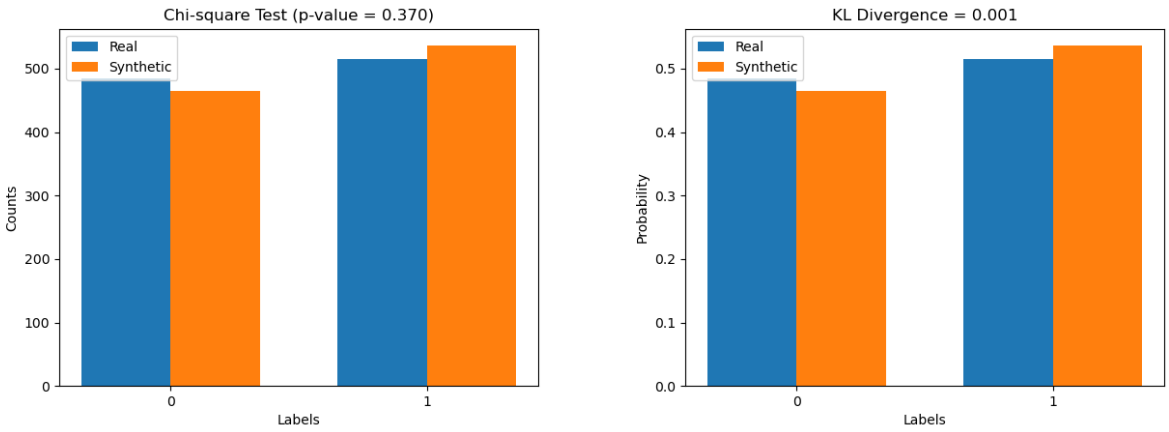

Fig. 6. Chi-square and KL divergence tests to compare the distributions of the Malaria and synthetic data. For both cases the p-value is statistically significant.

ACKNOWLEDGMENT

AI was partially supported by Project Tech4You - Technologies for climate change adaptation and quality of life improvement, n. ECS0000009 (Directorial Decree n. 2021/3277). We thank the eXtended Reality & Artificial Intelligence Lab (XR&AI Lab @ University of Basilicata) for algorithms development support.

AUTHOR CONTRIBUTIONS

The authors contributed equally to this work.

REFERENCES

[1] Martín Abadi et al. "TensorFlow: a system for Large-Scale machine learning". In: *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. 2016, pp. 265–283.

[2] Richard J Chen et al. "Synthetic data in machine learning for medicine and healthcare". In: *Nature Biomedical Engineering* 5.6 (2021), pp. 493–497.

[3] Yuheng Fan et al. "A survey of emerging applications of diffusion probabilistic models in mri". In: *Meta-Radiology* (2024), p. 100082.

[4] Charles Harris et al. "Flexible Small-Molecule Design and Optimization with Equivariant Diffusion Models". In: *ICLR 2023-Machine Learning for Drug Discovery workshop*. 2023.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

[6] *Imagenet Dataset*. https://www.image-net.org/index.php. Accessed: 2024-05-28.

[7] John B Ingraham et al. "Illuminating protein space with a programmable generative model". In: *Nature* 623.7989 (2023), pp. 1070–1078.

[8] Firas Khader et al. "Denoising diffusion probabilistic models for 3D medical image generation". In: *Scientific Reports* 13.1 (2023), p. 7303.

[9] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[10] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[11] *malaria Dataset*. https://ceb.nlm.nih.gov/repositories/malaria-datasets/. Accessed: 2024-05-28.

[12] Gilda Manfredi et al. "Treesketchnet: From sketch to 3d tree parameters generation". In: *ACM Transactions on Intelligent Systems and Technology* 14.3 (2023), pp. 1–29.

[13] Adam Paszke et al. "Automatic differentiation in PyTorch". In: (2017).

[14] Sivaramakrishnan Rajaraman et al. "Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images". In: *PeerJ* 6 (2018), e4568.

[15] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[16] Arne Schneuing et al. "Structure-based drug design with equivariant diffusion models". In: *arXiv preprint arXiv:2210.13695* (2022).

[17] Jascha Sohl-Dickstein et al. "Deep unsupervised learning using nonequilibrium thermodynamics". In: *International conference on machine learning*. PMLR. 2015, pp. 2256–2265.

[18] Yang Song and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution". In: *Advances in neural information processing systems* 32 (2019).

[19] Brian L Trippe et al. "Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem". In: *arXiv preprint arXiv:2206.04119* (2022).

[20] Allan Tucker et al. “Generating high-fidelity synthetic patient data for assessing machine learning healthcare software”. In: *NPJ digital medicine* 3.1 (2020), pp. 1–13.

[21] Ling Yang et al. “Diffusion models: A comprehensive survey of methods and applications”. In: *ACM Computing Surveys* 56.4 (2023), pp. 1–39.