DATA AUGMENTATION FOR PALM KERNEL OIL EXTRACTION MACHINE MODEL DEVELOPMENT USING GENERATIVE ADVERSARIAL NETWORK (GAN) APPROACH

Emmanuel Udama Odeh[1]
Department of Mechanical and Aerospace Engineering
University of Uyo, Akwa Ibom State, Nigeria
emmanuelodeh@uniuyo.edu.ng

Emem Sunday Ezekiel[2]
Department of Mechanical and Aerospace Engineering
University of Uyo, Akwa Ibom State, Nigeria
ememezekiel@gmail.com

OLALEYE, O. Olukayode[3]
Marine Engineering Department,
Maritime Academy, Oron, Akwa Ibom State, Nigeria
kayola_nan@yahoo.com

***Abstract*— In this work, data augmentation for palm kernel oil (PKO) extraction machine model development using Generative Adversarial Network (GAN) approach is presented. The essence of the study is to address the problem of paucity of data in situations where empirical data collection is expensive for data-driven modelling; a situation that is prevalent in many industrial setups. The case study Palm Kernel Oil (PKO) extraction machine has only 125 data records whereas over 1000 data records are required for effective machine learning modeling of the machine's performance. The parameters of the PKO extractor machine considered are moisture content, oil yield, shaft speed and cone gap. The results show that the original dataset (with 125 data records) and the GAN augmented dataset (with 5000 data records) have the same correlation results with shaft speed having the highest correlation coefficient of 0.71 with respect to the oil yield in both datasets (the original and augmented datasets), the moisture content has correlation coefficient of -0.11 with respect to the oil yield in both datasets, while the cone gap has the least correlation coefficient of -0.056 with respect to the oil yield in both datasets. Also, at 95 % confidence level there is no significant difference between the mean of the original and augmented datasets for each of the four parameters since the confidence interval of the augmented dataset contains the mean of the original dataset for each of the four parameters.**

Keywords*— *Data Augmentation, Palm Kernel Oil Extraction Machine, Machine Learning Model, Generative Adversarial Network, Optimization Model

1. Introduction

Nowadays, there is growing adoption of precision and smart technologies in virtually every discipline [1,2,3]. The precision and smart concepts have also permeated the industrial sector where data driven approaches are used in conjunction with artificial intelligent models to characterize the behavior and operations of industrial machines and systems [4,5,6]. In many cases, the needed data for studying a given case study machine is grossly insufficient requiring that additional data must either be acquired through empirical means or through data augmentation [7,8,9]. In the cases where empirical data collection is expensive, data augmentation becomes the best option [10,11].

In data augmentation, the sample original dataset is employed to generate additional data items that maintains the same statistical features are the original dataset [12,13]. There are several ways that such synthetic data can be generated however the performance of each approach can be assessed using different statistical measures [14,15]. The statistical parameters like mean, standard deviation, confidence interval, among other features are used and the goal is to ensure that the synthesized data records accurately maintains the pattern of the original dataset.

Accordingly, in this study, the data records empirically acquired for a 10-ton palm oil extractor machine is studied. The 125-record dataset was used in a Generative Adversarial Network (GAN) model to generate thousands of additional data records and the results are evaluated to assess the effectiveness of the GAN model is replicating the original data records of the PKO extractor machine [16,17]. The results obtained in such study is relevant in machine learning modeling of the PKO extractor machine with focus on prediction and optimization of the oil yield of such machine.

2. Methodology

In this work, data augmentation is carried out for palm kernel oil extraction machine using available data on four parameters empirically acquired from the case study machine. The parameters considered are moisture content, oil yield, shaft speed and cone gap. The available dataset has about 125 rows of data with each of the four parameters present. However, for machine learning model, several hundreds of data records are required. This is because:

i. The machine learning models require a significant amount of data to learn complex patterns and avoid overfitting.

ii. With 125 rows, splitting into training and validation sets would leave very few samples for learning and testing.

So, this study presents an approach to conduct the required data augmentation to generate additional data records which maintain the same pattern as the original dataset. Specifically, Generative Adversarial Networks (GANs) is used to augment the dataset by generating synthetic data that follows the same distribution as the original data.

2.1 How Generative Adversarial Networks (GANs) Work

The GAN model architecture is shown in Figure 1. The architecture shows that the GAN model has a data sample Generator and data sample Discriminator. The Generator is train with the case study data records such that it can randomly generate sample data records that fits the patterns present in the actual data records. At the same time, the Discriminator tries to identify the data sample generated by the Generator as fake. At the initial time, the Discriminator will easily identify the generated data records as fake but as the training continues, the Generator masters the data pattern and then generates random data samples which cannot be identified as fake by the Discriminator. In this way, the GAN model can be used to generate the required data records which are good representation of the machine operation settings without being empirically measured.

i. A GAN consists of two neural networks:
   a) Generator (G): Creates synthetic data points.
   b) Discriminator (D): Distinguishes between real and synthetic data.

ii. They are trained in a min-max game:
   a) The Generator tries to produce realistic data.
   b) The Discriminator tries to identify whether the data is real or synthetic.

iii. The training continues until the Discriminator can no longer distinguish between real and synthetic data, ensuring high-quality synthetic data generation.

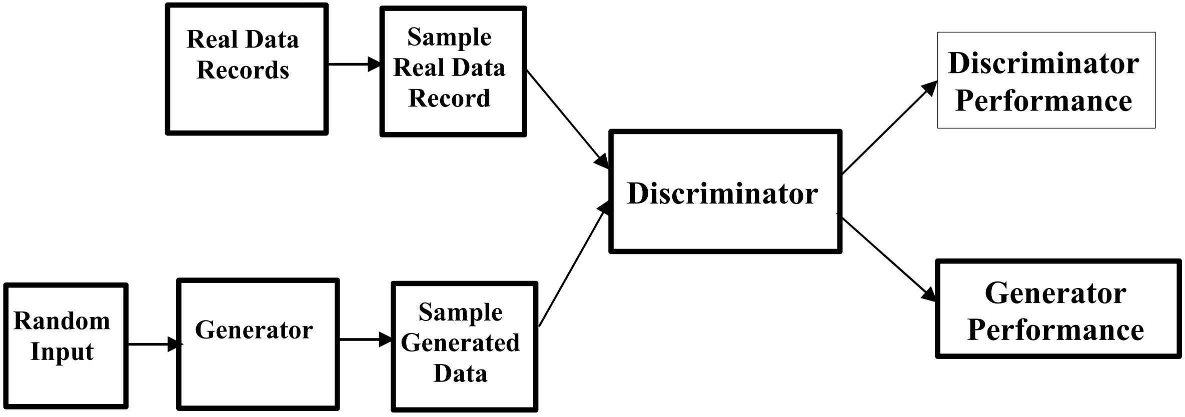

Figure 1 The GAN model architecture

2.2 Analytical procedure employed in the GAN for Data Augmentation

a) **Step 1: Data Preparation**

i. Normalize the data to a range suitable for neural networks (e.g., [0, 1]).

ii. Arrange the dataset with input features:
   a) Main Shaft Speed (rpm)
   b) Cone Gap (mm)
   c) Moisture Content (%)

iii. Output feature: **Oil Yield**

b) **Step 2: GAN Architecture**

i. **Generator (G):**
   a) Input: Random noise vector *z* sampled from a Gaussian distribution.
   b) Output: Synthetic data which maintains the same pattern as the original dataset.
   c) Activation Function: **ReLU** for hidden layers, **Tanh** for output layer.

ii. **Discriminator (D):**
   a) Input: Real or synthetic data point.
   b) Output: Probability of the data point being real.
   c) Activation Function**: Leaky ReLU** for hidden layers, **Sigmoid** for output layer.

c) **Step 3: Loss Functions and Training**

i. **Generator Loss (G-Loss):**

$$L_G = -\log\Big(D\big(G(z)\big)\Big) \quad (1)$$

This maximizes the probability of the **Discriminator** classifying the synthetic data as real.

ii. **Discriminator Loss (D-Loss):**

$$L_D = -\left[\log(D(x)) + \log\left(1 - D\left(G(z)\right)\right)\right] \qquad (2)$$

This minimizes the probability of misclassification.

iii. **Training Process:**

a) **Step 1:** Train the **Discriminator** on real and synthetic data.

b) **Step 2:** Train the **Generator** to produce more realistic synthetic data.

c) Repeat until the **Discriminator** can't distinguish real from synthetic data.

d) **Step 4: Data Augmentation and Evaluation**

i. Generate synthetic samples until the dataset size is significantly increased (e.g., from 125 to 5000 rows).

ii. Combine the synthetic data with the original data for training the machine learning models.

iii. **Validate** the augmented dataset using:

a) **t-SNE** visualization to check for realistic distribution overlap between real and synthetic data.

b) **Statistical tests** (e.g., KS Test) to ensure the augmented data follows the same distribution as the original data.

2.3 Implementation of GAN in Python

The models presented in the Equations 1 and 2 were implemented. The following hyperparameters were tuned and their impacts are discussed below:

Noise Dimension: The is part of the input which represents the latent space. It introduces randomness and diversity in the generated data. A larger noise dimension provides more variation, potentially generating more diverse samples. However, if too large, it might lead to overfitting or difficulty in learning a smooth mapping from noise to data space. Conversely, a small value might restrict diversity, leading to mode collapse (where the generator produces limited variations). In this work, a noise dimension of 100 was selected.

Learning Rate: It controls the step size for updating model weights during backpropagation. It also balances the speed and stability of training. If the learning rate is too high, the model might overshoot the optimal solution, leading to instability and failure to converge. If too low, the model may converge slowly or get stuck in local minima. The generator is trained at a low learning rate while the discriminator has slightly higher learning rate. This ensures the discriminator does not become too powerful too quickly, allowing the generator to learn effectively. In this work, the generator learning rate of $1e^{-4}$ was selected while $1e^{-8}$ was selected for discriminator.

Batch Size: The batch size impacts gradient estimation and model stability. In this work, batch size of 64 was selected.

Label Smoothing: Instead of using 1 for real and 0 for fake, labels are smoothed to 0.9 for real data. This prevents the discriminator from becoming too confident and reduces the risk of overfitting in the discriminator. Makes the GAN training more stable by preventing the discriminator from overpowering the generator. In this work 0.9 was selected for label smoothening.

Number of Epochs: In this work, 10000 epochs was selected. The summary of the GAN model's hyperparameters and their values are presented in Table 1.

Table 1: The summary of the GAN model's hyperparameters and their values

Hyperparameter	Value
Noise dimension	100
Discriminator learning rate	$1e^{-8}$
Generator learning rate	$1e^{-4}$
Batch size	64
Label smoothening	0.9
Number of Epochs	10000

3. Results and Discussion

The results in Table 2 and Figure 2 are for the GAN model. The results in Table 2 and Figure 2 show that between the $1^{st} - 1000^{th}$ epochs, the generator struggled to fool the discriminator on synthetic data. However, from 2000^{th} and above (after sufficient training) the negative value trend on the generator loss shows that the generator got too confident in fooling the discriminator on synthetic data.

Table 2: Discriminator losses and Generator losses

Epoch	Discriminator loss	Generator loss
0	8.6726	0.0639
1000	20.4780	-12.1951
2000	85.2671	-76.3618
3000	227.6830	-220.3845
4000	468.8762	-464.8153
5000	816.1927	-819.8204
6000	1305.7658	-1289.1628
7000	1922.7521	-1906.0122
8000	2627.3137	-2581.6962
9000	3558.2502	-3467.1650

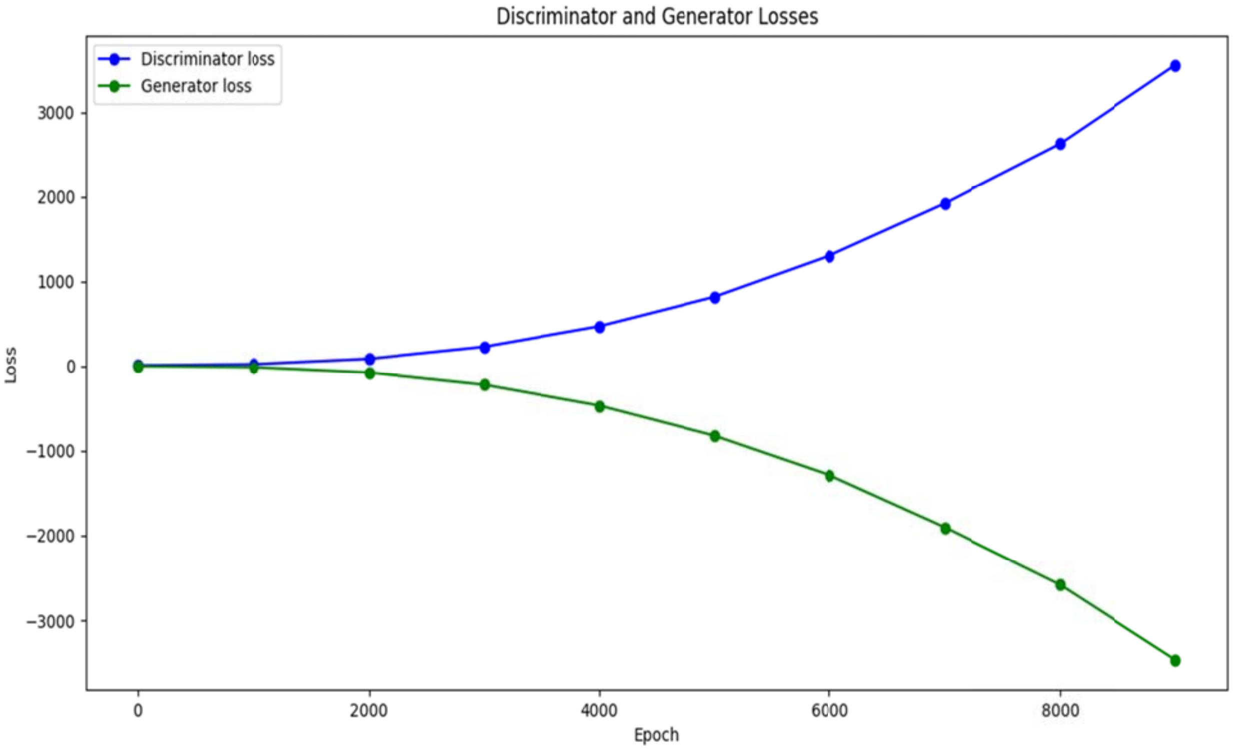

Figure 2: The line chart of the GAN model discriminator losses and generator losses

3.1 The results of the correlation matrix and descriptive statics of the original and augmented dataset

The correlation matrix of the original data is shown in Figure 3 and Table 3 while the correlation matrix of the augmented data is shown augmented dataset is presented in Figure 4.8 and Table 4. The summary of descriptive statics of the original and GAN augmented dataset is presented in Table 5. The results show that the original and the GAN augmented dataset have the same correlation results with shaft speed having the highest correlation coefficient of 0.71 with respect to the oil yield in both datasets, the moisture content has correlation coefficient of -0.11 with respect to the oil yield in both datasets, while the cone gap has the least correlation coefficient of -0.056 with respect to the oil yield in both datasets.

Table 3: Correlation matrix for the original dataset

	Shaft speed	**Cone gap**	**Moisture content**	**Oil yield**
Shaft speed	1.0000	-6e-16	5e-17	0.71
Cone gap	-6e-16	1.0000	-6.4e-17	-0.056
Moisture content	5e-17	-6.4e-17	1.0000	-0.11
Oil yield	0.71	-0.056	-0.11	1.0000

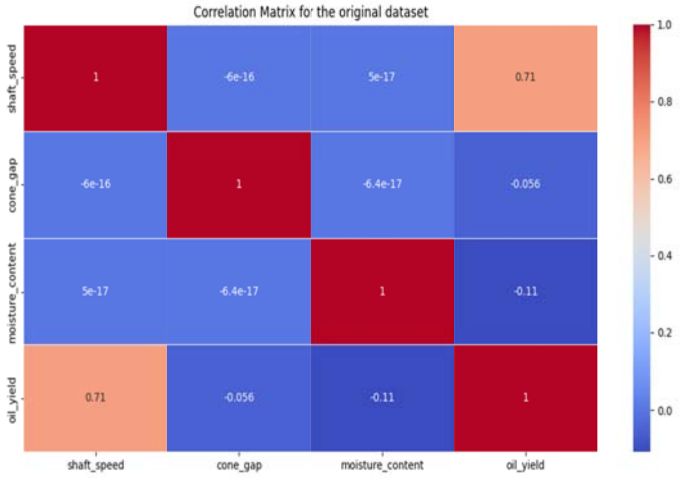

Figure 3: Correlation matrix for original data

Table 4: Correlation matrix for the augmented dataset

	Shaft speed	**Cone gap**	**Moisture content**	**Oil yield**
Shaft speed	1.0000	-6e-16	5e-17	0.71
Cone gap	-6e-16	1.0000	-6.4e-17	-0.056
Moisture content	5e-17	-6.4e-17	1.0000	-0.11
Oil yield	0.71	-0.056	-0.11	1.0000

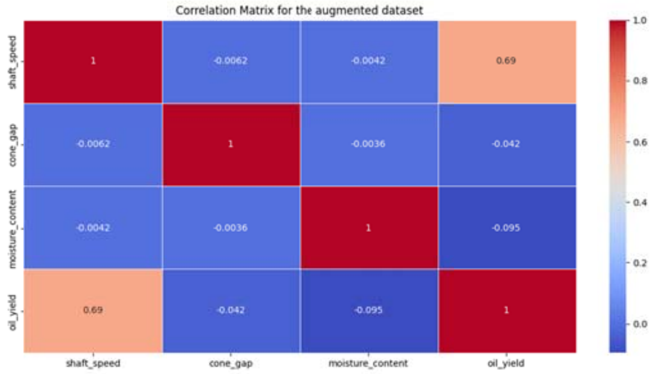

Figure 4: correlation matrix for the augmented dataset

Table 5 The summary of descriptive statics of the original and GAN augmented dataset

Statistics	Original Dataset	Augmented Dataset	Percentage Difference
SHAFT SPEED			
Mean	18	18.03	0.17%
Standard deviation	2.83	2.82	0.35%
Min	14	14	0.00%
Max	22	22	0.00%
CONE GAP			
Mean	1.5	1.502	0.13%
Standard deviation	0.71	0.71	0.00%
Min	0.5	0.5	0.00%
Max	2.5	2.5	0.00%
MOISTURE CONTENT			
Mean	10	10.02	0.20%
Standard deviation	2.83	2.82	0.35%
Min	6	6	0.00%
Max	14	14	0.00%
OIL YIELD			
Mean	38.87	38.9	0.08%
Standard deviation	2.46	2.39	2.85%
Min	33.8	33.8	0.00%
Max	43.4	43.4	0.00%

3.2 Calculation of the Confidence Interval (CI) for both the original data and the augmented dataset

Calculation of the Confidence Interval (CI) for the shaft speed was conducted using the statskingdom.com online tool available at: https://www.statskingdom.com/confidence-interval-calculator.html. The following parameters obtained from Table 5 were used for the CI computation;

i. The population standard deviation (σ)= 2.84

ii. Sample mean ($\bar{x}$) = mean of augmented dataset = 18.03

iii. Sample size (n) = size of augmented dataset = 5000

iv. The confidence level (CL) – 95 %

The steps used for the CI computation are as follows:

i. Calculate the significance level (α):

- $\alpha = 1 - CL = 1 - 0.95 = 0.05.$

ii. Calculate the probability (p):

- $p= 1 - \alpha/2 = 1 - 0.05/2 = 0.975.$

iii. Calculate the z-score ($Z_{(p)}$):

- $Z_{(p)} = Z_{(1-\alpha/2)} = Z_{0.975} = 1.96$

iv. Calculate the Confidence Interval (CI)

- $CI = \bar{x} \pm Z_{(1-\alpha/2)} \left(\frac{\mu}{\sqrt{n}}\right) = 18.03 \pm 1.96 \left(\frac{2.84}{\sqrt{5000}}\right)$
- $CI = 18.3 \pm 1.96\,(0.03479) = 18.3 \pm 0.06819$
- $CI = 17.96181\,, 18.09819$

- "the interpretation is that with 95% confidence the population mean is between 17.8 and 18.3, based on 500 samples."

Essentially, for the Shaft Speed, since the Confidence Interval (CI) for the shaft speed at 95% confidence level contains the mean of the original mean which is 18, it means that the original dataset and the GAN augmented dataset have the same mean. This justified the use of the augmented shaft speed data for machine learning model training and validation.

3.3 Summery of the results for the Confidence Interval (CI) the four parameters

In this study the original dataset is the reference dataset. The mean of the reference dataset is then used to check the outcome of the confidence interval for the systemized dataset.

3.3.1 Shaft Speed

i. Mean of reference dataset: 18.000000

ii. Original 95% CI: (np.float64(17.497262347650857), np.float64(18.502737652349143))

iii. Augmented 95% CI: (np.float64(18.28712460629906), np.float64(18.44247539370094))

iv. t-Test: t-statistic = -1.437436354302596, p-value = 0.15065511885792557

v. The interpretation: At 95 % confidence level there is no significant difference between the mean of the original and augmented data on Shaft Speed since the

CI of the augmented dataset contains the mean of the original dataset.

3.3.2 Cone Gap

i. Mean of reference dataset: 1.500000
ii. Original 95% CI: (np.float64(1.3743155869127142), np.float64(1.6256844130872858))
iii. Augmented 95% CI: (np.float64(1.4997274536735585), np.float64(1.5388725463264417))
iv. t-Test: t-statistic = -0.30186424514155114, p-value = 0.7627678062512935
v. The interpretation: At 95 % confidence level there is no significant difference between the mean of the original and augmented data on Cone Gap since the CI of the augmented dataset contains the mean of the original dataset.

3.3.3 Moisture Content

i. Mean of reference dataset: 10.000000
ii. Original 95% CI: (np.float64(9.497262347650857), np.float64(10.502737652349143))
iii. Augmented 95% CI: (np.float64(9.942439498979542), np.float64(10.099160501020457))
iv. t-Test: t-statistic = -0.0812601013849461, p-value = 0.9352382777896682
v. The interpretation: At 95 % confidence level there is no significant difference between the mean of the original and augmented data on Moisture Content since the CI of the augmented dataset contains the mean of the original dataset.

3.3.4 Oil Yield

i. Mean of reference dataset: 38.879200
ii. Original 95% CI: (np.float64(38.443576759585724), np.float64(39.31482324041427))
iii. Augmented 95% CI: (np.float64(39.08372920353116), np.float64(39.21639079646883))
iv. t-Test: t-statistic = -1.249358713788711, p-value = 0.21159097979058905
v. The interpretation: At 95 % confidence level there is no significant difference between the mean of the original and augmented data on Oil Yield since the CI of the augmented dataset contains the mean of the original dataset.

4. Conclusion

An approach to synthetically generate additional data items based on available original data is presented. In this case, the Generative Adversarial Network (GAN) approach is used to generate additional data for a case study palm kernel oil extraction machine. The study is conducted due to the paucity of original dataset for carrying out machine learning-based modeling of the machine performance. Hence, the GAN model was used to generate a dataset with about 5000 data records which maintained the same pattern and arithmetic mean as the original 125 data records. This approach is cost effective when compared with empirical data collection for different machine configurations that will usually require materials, time and labour which together amounts to high cost of data collection.

References

1. Mhlongo, S., Mbatha, K., Ramatsetse, B., & Dlamini, R. (2023). Challenges, opportunities, and prospects of adopting and using smart digital technologies in learning environments: An iterative review. *Heliyon*, *9*(6).
2. Mittal, S., Khan, M. A., Romero, D., & Wuest, T. (2019). Smart manufacturing: Characteristics, technologies and enabling factors. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *233*(5), 1342-1361.
3. Oosthuizen, R. M. (2022). The fourth industrial revolution–smart technology, artificial intelligence, robotics and algorithms: Industrial psychologists in future workplaces. *Frontiers in artificial intelligence*, *5*, 913168.
4. Trakadas, P., Simoens, P., Gkonis, P., Sarakis, L., Angelopoulos, A., Ramallo-González, A. P., ... & Karkazis, P. (2020). An artificial intelligence-based collaboration approach in industrial iot manufacturing: Key concepts, architectural extensions and potential applications. *Sensors*, *20*(19), 5480.
5. Peres, R. S., Jia, X., Lee, J., Sun, K., Colombo, A. W., & Barata, J. (2020). Industrial artificial intelligence in industry 4.0-systematic review, challenges and outlook. *IEEE access*, *8*, 220121-220139.
6. Wan, J., Li, X., Dai, H. N., Kusiak, A., Martinez-Garcia, M., & Li, D. (2020). Artificial-intelligence-driven customized manufacturing factory: key technologies, applications, and challenges. *Proceedings of the IEEE*, *109*(4), 377-398.
7. Ding, J., Li, X., Kang, X., & Gudivada, V. N. (2019). A case study of the augmentation and evaluation of training data for deep learning. *Journal of Data and Information Quality (JDIQ)*, *11*(4), 1-22.

8. Bansal, M. A., Sharma, D. R., & Kathuria, D. M. (2022). A systematic review on data scarcity problem in deep learning: solution and applications. *ACM Computing Surveys (Csur)*, *54*(10s), 1-29.
9. Lemley, J., Bazrafkan, S., & Corcoran, P. (2017). Smart augmentation learning an optimal data augmentation strategy. *Ieee Access*, *5*, 5858-5869.
10. Mumuni, A., & Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. *Array*, *16*, 100258.
11. Nanni, L., Paci, M., Brahnam, S., & Lumini, A. (2021). Comparison of different image data augmentation approaches. *Journal of imaging*, *7*(12), 254.
12. Fazekas, B., & Kiss, A. (2018, August). Statistical data generation using sample data. In *European Conference on Advances in Databases and Information Systems* (pp. 29-36). Cham: Springer International Publishing.
13. Lu, Y., Shen, M., Wang, H., Wang, X., van Rechem, C., Fu, T., & Wei, W. (2023). Machine learning for synthetic data generation: a review. *arXiv preprint arXiv:2302.04062*.
14. Figueira, A., & Vaz, B. (2022). Survey on synthetic data generation, evaluation methods and GANs. *Mathematics*, *10*(15), 2733.
15. Alaa, A., Van Breugel, B., Saveliev, E. S., & Van Der Schaar, M. (2022, June). How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International conference on machine learning* (pp. 290-306). PMLR.
16. Antoniou, A., Storkey, A., & Edwards, H. (2017). Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*.
17. Biswas, A., Md Abdullah Al, N., Imran, A., Sejuty, A. T., Fairooz, F., Puppala, S., & Talukder, S. (2023). Generative adversarial networks for data augmentation. In *Data Driven Approaches on Medical Imaging* (pp. 159-177). Cham: Springer Nature Switzerland.