Journal of Computer Science

Original Research Paper

A Multi-Split Cross-Strategy for Enhancing Machine Learning Algorithms Prediction Results with Data Generated by Conditional Generative Adversarial Network

[1]Abdelfattah Abassi, [1,2]Brahim Bakkas, [1,3]Mostapha El Jai, [1]Ahmed Arid and [1]Hussain Benazza

[1]Department of Computer Science, Ecole Nationale Supérieure d'Arts et Métiers (ENSAM-MEKNES), Moulay Ismail University, Meknes, Morocco
[2]Department of Computer Science, Regional Center for Teaching and Training Professions, Meknes, Morocco
[3]Euromed Center of Research, Euromed Polytechnic School, Euromed University, FEZ, Morocco

Article history
Received: 07-02-2024
Revised: 15-03-2024
Accepted: 22-03-2024

Corresponding Author:
Abdelfattah Abassi
Department of Computer Science, Ecole Nationale Supérieure d'Arts et Métiers (ENSAM-MEKNES), Moulay Ismail University, Meknes, Morocco
Email: abassi.mri@gmail.com

Abstract: In this study, we present a Multi-Split Cross-Strategy (MSC-Strategy) designed to leverage synthetic tabular data generated by a Conditional Generative Adversarial Network (CGAN). Our study aims to investigate the potential of synthetic data in comparison to real-world data for improving machine learning predictive results. Firstly, we develop a CGAN architecture tailored to generate synthetic tabular data, trained on a comprehensive real-world dataset. Secondly, we validate the synthetic data generated by the CGAN to ensure its statistical fidelity and resemblance to the distribution of real data. Finally, we selectively leverage a subset of the generated data and apply our strategy to create a new combined training set comprising the training set of real data and the chosen subset of generated data. To validate our approach, we employ six diverse regression models: Decision Tree (DT), K-Nearest Neighbors (KNN), Random Forest (RF), XGB Regressor (XGB), and Support Vector Regressor (SVR). Each model is trained and tested using a training set of real data, generated data, combined data (training set of real data and generated data), and data formed by our MSC strategy. Our findings indicate that the training set formed by our MSC strategy demonstrates remarkable predictive performance compared to real-world data and generated data, highlighting its ability to enhance the prediction of machine learning models using only a subset of generated data.

Keywords: Conditional Generative Adversarial Networks, Tabular Data Generation, Machine Learning

Introduction

One major obstacle that frequently comes in the way of data scientists trying to find answers to principal issues is the lack of sufficient data. Some machine learning models require kinds or amounts of data, which might not be available (Soori *et al*., 2023). High financial costs, time constraints, or engagement issues like privacy, safety, or time investment can all contribute to this limitation (Calderaro, 2015). It becomes impractical or perhaps impossible to obtain more data under such conditions.

The adoption of synthetic data is becoming a feasible solution to this problem (Alloza *et al*., 2023; El Emam *et al*., 2020; Ladeira Marques *et al*., 2020). When data is created using artificial processes as opposed to being gathered from real-world events, it is referred to as synthetic data. Synthetic augmentation of data quantity and variety can result in significant improvements in performance for machine-learning models when executed right (Hernandez *et al*., 2022). The graph provided by Gartner illustrates the expected rise in the use of synthetic data in machine-learning applications in the upcoming years (Chatterjee and Byun, 2023).

Applications for synthetic data's scalability may be found in several sectors, including robots, finance, security, and autonomous cars (Rajotte *et al*., 2022). Its importance, however, there are many obstacles in the way of collecting large amounts of data (Saxena and Cao, 2021). Data

collection suffers from the high expenses of linked activities as well as the difficulty in finding samples who meet certain criteria. As a result, investigating artificial methods for producing verifiable data becomes an important task. A variety of techniques can be utilized to produce artificial data, but one particularly interesting approach is the application of GANs. GAN is composed of two neural networks that participate in an adversarial process. The discriminator attempts to separate synthetic data from actual data and the generating network attempts to fool it. The generator specifically aims to generate information that closely approximates actual data. The generator is penalized if the discriminator is successful in identifying synthetic information and vice versa. The model gains the ability to produce data that shows similarities to the original dataset through this iterative process. In-depth information on the architecture of these models is provided by Ian Goodfellow's groundbreaking paper (Goodfellow *et al*., 2020).

In recent years, image learning has been the principal field in which GANs are applied. Examples of these applications include image synthesis, colorization, upscaling, and restoration (Pan *et al*., 2019; Sorin *et al*., 2020; Lu *et al*., 2022). Thispersondoesnotexist.com is a well-known example, showcasing a GAN trained on actual faces that can produce realistic-looking but wholly fake faces. However, using picture data to train GAN models can be an exhausting procedure that takes weeks or months at times.

Despite this, text and numerical data have also been effectively processed using GANs (Alqahtani *et al*., 2021). Remarkably, these models need much less training time when applied to non-image data. Because of their lower acceptance difficulties, GANs are becoming a more useful tool in data science and artificial intelligence research. The creation of conditional tabular data (CGAN) is one particular use for GAN.

CGAN has been developed for the synthesis of tabular data. The purpose of CGAN, an extension of GANs designed especially for tabular data, is to produce synthetic data with characteristics similar to those of the original data. Information arranged in a table or spreadsheet format, usually with rows and columns, is referred to as tabular data. Insaf Ashrapov's article "Tabular GANs for Uneven Distribution" (Ashrapov, 2020) explored the idea of using CGANs to generate tabular data from real tabular data.

However, using generated data to improve machine learning algorithm prediction error is not guaranteed. To address this challenge, this study proposes an MSC strategy to use a subset of generated data combined with original data to train machine-learning algorithms. For this purpose, we split the original data into training and testing sets. The training data is used to train CGAN to generate synthetic data, while the testing data is used to evaluate the prediction performance of machine learning algorithms using both original data, generated data, combined data, and data formed by our MSG strategy. Our MSC strategy utilizes different numbers of splits to divide the generated data into sub-sets. We demonstrate through simulations that our MSC strategy improves the performance of machine learning algorithms compared to using real data alone, generated data alone, or simply combining all available data.

Materials and Methods

Conditional Generative Adversarial Network

A modified version of GANs that introduces the idea of conditioning the generative process on extra information is Conditional Generative Adversarial Networks (CGANs). While CGANs enable the production of samples conditioned on input data, ordinary GANs generate samples from random noise. Figure 1 illustrates a basic depiction of the GANs and CGANs model's structure.

The CGANs are composed by:

- Generator: CGANs are made up of a generator network that aims to produce realistic samples, just like GANs. In CGANs, on the other hand, the generator generates samples (y) using both conditional information and random noise (z) as input
- Discriminator: The discriminator's job is still to tell the difference between created and true samples. It examines the given conditions in addition to the generated and actual samples. Conditional Data Entry: Which can be any kind of extra data that directs the generating process, is introduced by CGANs. Depending on the application, this could comprise class labels, attributes, or any other related information ($_{real}$)

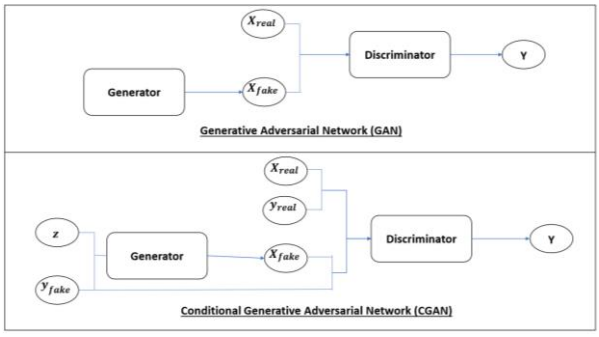

Fig. 1: Conditional and generative adversarial networks

MSC-Strategy Coupled with CGAN

In this section, we detail our strategy for enhancing the performance of machine learning models. As discussed in the preceding section, we partition the real data into training and test sets. The training set serves as input for training the CGAN, while the test set is employed to evaluate the performance of each machine-learning model discussed in this study. Only the training set of original data and the entire set of generated data are used for the initial test. The test set from the actual dataset is then used to conclude the performance evaluation. During the testing phase, it becomes evident that the size of the generated data used as a training set directly impacts the performance of each model. Some models exhibit improved performance when trained solely on real data, without incorporating the fully generated data. Conversely, other models demonstrate enhancement only when a combination of the training set from real data and a portion of generated data is utilized. Interestingly, the same combination that proves effective for one model may yield poor performance for another model. This simulation leads us to assert that different portions of generated data can be effectively utilized in conjunction with the training set of real data to enhance the performance of a specific model, while a different combination may be more suitable for improving another model.

Based on these observations, we conclude that we can divide the generated data into k-sub-data. We then form a new training set by combining each k-sub-data with the training set of original data. We then test the model's performance with this new training set. Each combination has a different prediction error and we then choose the combination with the minimum error. It is noted that the number of splits k selected has an influence on our strategy. For example, if we choose $k = 1$, in this case, we return to the initial case in which we combine the train data of real data with the generated data during the initial phase. As a result, k is a hyperparameter of our strategy. Figure 2 presents the main steps of the MSC strategy and Algorithm 1 describes the steps to select the best Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) metrics for a given machine learning model.

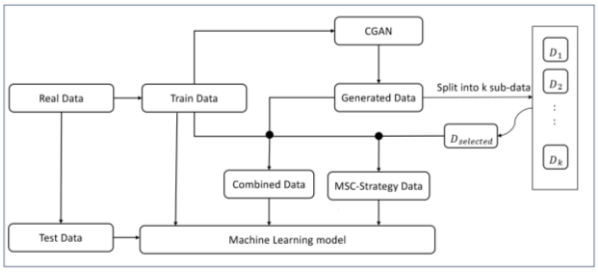

Fig. 2: The main steps of the MSC-strategy

Algorithm 1 MSC-Strategy algorithm

function GETBESTRMSEANDMAE
  Input: model,X_{train}, y_{train}, X_{test} , y_{test} , X_g, y_g, k
  Output: *bestRMSE, bestMAE*,
  bestsplitRMSE, bestsplitMAE
  Initialisation:
  bestRMSE ← ∞
  bestMAE ← ∞
  bestsplitRMSE ← 0
  bestsplitMAE ← 0
  D_1 ← Splits X_g into k parts
  D_2 ← Splits y_g into k parts
  for D_{1i}, D_{2i}**in**{D_1, D_2} **do**
    X ← X_{train} ∪ D_{1i}
    Y ← y_{train} ∪ D_{2i}
    Train (model, X, Y)
    RMSE, MAE ← Evalate(model, X_{test} , y_{test})
    if (RMSE < bestRMSE) **then**
      bestRMSE ← RMSE
      bestsplitRMSE ← i
    end if
    if (MAE < bestMAE) **then**
      bestMAE ← MAE
      bestsplitMAE ← i
    end if
  end for
  return bestRMSE, bestsplitRMSE, bestMAE, bestsplitMAE
end function

Here's a step-by-step description:

- Input Parameters:
  - Model: The machine learning model under evaluation
  - X_{train}, y_{train} Training data
  - X_{test}, y_{test}: Test data
  - X_g,y_g: Generated data by CGAN
  - k: The number of splits used for dividing the generated data
- Initialization:
  - Initialize bestRMSE and bestMAE to infinity. These variables are updated during the execution of the algorithm to store the best metrics found
  - Data splitting:
  - Split the generated data X_g into k parts, stored in D_1
  - Split the corresponding labels y_g into k parts, stored in D_2
- Model evaluation loop:
  - Iterate over each pair of splits D_{1i} and D_{2i} from the generated data
  - Form a new training set by combining the original training set with the current split (D_{1i}, D_{2i})

  - Train the model with the combined training set
  - Evaluate the trained model on the test set (X_{test}, y_{test}) to obtain RMSE and MAE metrics

- Update best metrics:

  - If the RMSE and MAE obtained from the current model are both better (lower) than the current best metrics, update bestRMSE and bestMAE
  - After evaluating the model for all splits, return the best RMSE, MAE, and best-split index

To determine which model performs best on RMSE and MAE measures, the program iteratively trains and evaluates the model on various combinations of the generated data.

Results and Discussion

In this section, we discuss the quality of the generated tabular data and the performance of the models presented in this study on real data, generated data, a combined dataset, and data formed by the MSC strategy. The data used in this study contains historical tabular power consumption data collected from Tetouan City, situated in the northern region of Morocco. The dataset is unique and complete, with no missing data, collected at 10 min intervals, spanning from January 1, 2017, 00:00:00 to December 31, 2017, 23:50:00. It contains records of date, time, and consumption figures for the three distribution networks (Salam and El Hibaoui, 2018). This dataset contains four features that have already been presented, namely: Temperature (T), Humidity (H), Wind Speed (WS), General Diffuse Flows (GDF), Diffuse Flows (DF), and other features created from a datetime column such as hour (h), day (d), minute (m), month (mon), day of the year (dy) and quarter (q). The target variable of this dataset is energy consumption. The presented work was then turned into a Python package called tabgan, which makes it easier to use CGAN to produce tabular data. The creation of numerical and textual category data is supported by the library. Table 1 presents the parameters of our CGAN architecture.

Also, Table 2 presents the key hyperparameters employed for various regression models in the study. Noteworthy, parameters are specified for each model, providing transparency and facilitating reproducibility of the experimental setup.

Quality of Generated Tabular Data

In this section, the evaluation of synthetic tabular data generated using CGAN is explored. To make sure that the synthetic data is representative of the underlying data distribution and can be utilized successfully for tasks like data analysis, machine learning models, and decision-making, it is important to evaluate the quality of the generated data. While high-quality synthetic data can preserve the statistical features and patterns of the original data, poorly created data may produce biased or fraudulent results (Krippendorff, 2009).

A variety of evaluation methods are used to determine the quality of the generated tabular data. These methods include feature distribution comparison, feature correlation analysis, data-driven metrics, visual inspection (Lanovaz and Hranchuk, 2021), and summary statistics calculation (Kenney, 1939). In addition, we discuss the importance of user input and regression performance in evaluating the usefulness of synthetic data generated by our approach.

A computation of summary statistics on the produced and original data using Python and Panda's package is explored (McKinney, 2015). Quick overviews of the distribution of the data are provided by summary statistics like mean, median, and standard deviation, which can aid in discovering possible differences between the two datasets. It's crucial to remember that summary statistics might not be able to fully capture every component of data quality, so additional assessment techniques should be used in combination for an extensive assessment. The distribution of the target variable is presented in the column (energy).

Table 1: CGAN parameters and values

Component	Parameters
Generator	Learning rate: 0.005 Optimizer: Adam Loss function: Binary Cross-Entropy (BCE) Number of layers: 2 hidden layers with Leaky ReLU activation Gradient penalty: 10 (encourages Lipschitz continuity, crucial for W1 distance)
Discriminator	Learning rate: 0.005 Optimizer: Adam Loss function: Binary Cross-Entropy (BCE) Number of layers: 2 hidden layers with Leaky ReLU activation Gradient penalty: 10 (encourages Lipschitz continuity, crucial for W1 distance)

Table 2: Regressor parameters and values

Regressor	Parameters	Values
GB	n estimators	100.00
	learning rate	0.10
	max _depth	3.00
XGB	objective	REG squared error
	random _state	50.00
	max _depth	6.00
	learning rate	0.08
	n estimators	500.00
DT	max _depth	5.00
	random _state	42.00
KNN	n neighbors	5.00
RF	n estimators	30.00
	max _depth	7.00
	max _features	Auto
	min _samples split	2.00
	min _samples leaf	1.00
	random _state	42.00
SVR	C	1.00
	Kernel	RBF

Tables 3-4 present a summary of statistics for the original data and generated data. Due to space constraints, only the statistics for some features such as T, H, WS, GDF, and DF are presented.

We can see from the comparison that the generated data's summary statistics and the original data's summary statistics are basically comparable. The close to means, standard deviations, and percentiles show that the CGAN was successful in capturing the primary statistical characteristics of the original data. This implies that the generated data reflects the properties of the original distribution and is of a reasonably high quality.

However, the number of records in the generated data (44,000) is significantly larger than that in the original data (35,118), which is due to the CGAN's ability to generate a more extensive dataset. This increase in the dataset size could be beneficial for certain applications, but it's essential to ensure that the additional samples remain representative of the original data distribution. The number of records generated by CGAN is significant, but it is beyond the scope of this study.

The minimum, first quartile, median, third quartile, and maximum of each feature and target variable in both datasets are presented in Figs. 3-4 respectively. From both figures, the distributions of humidity, month, day of the week, quarter, day of the year, and hour are roughly symmetrical. The distributions of other features are skewed, with a few outliers. Overall, the generated data showed a representative quality and was quite like the real data, despite small differences across various variables.

For a more detailed visualization of the distribution of each feature, histograms, and kernel density plots are presented in Fig. 5. Also, the Heatmap correlation between real data and generated data is presented in Fig. 6.

Table 3: Summary statistics for original data

	T	H	WS	GDF	DF	Energy
Count	35118.00	35118.00	35118.00	35118.00	35118.00	35118.00
Mean	18.83	68.21	1.96	183.81	74.88	32330.43
Std	5.78	15.57	2.35	265.10	123.99	7120.87
Min	3.25	11.34	0.05	0.00	0.01	13895.70
25%	14.45	58.18	0.08	0.06	0.12	26335.93
50%	18.83	69.83	0.09	5.20	4.60	32279.85
75%	22.87	81.40	4.92	324.58	100.40	37261.59
Max	39.78	94.80	6.48	1163.00	936.00	52204.40

Table 4: Summary statistics for generated data

	T	H	WS	GDF	DF	Energy
Count	44000.00	44000.00	44000.00	44000.00	44000.00	44000.00
Mean	19.01	67.33	1.94	220.18	98.49	32676.64
Std	6.07	16.07	2.32	274.31	143.81	7739.22
Min	4.45	16.01	0.06	0.01	0.04	13720.18
25%	14.56	57.29	0.08	0.07	0.14	26651.10
50%	18.81	69.36	0.14	73.63	37.63	32499.48
75%	23.02	80.60	4.91	406.42	140.02	37679.73
Max	36.86	93.70	4.97	937.65	748.00	55976.28

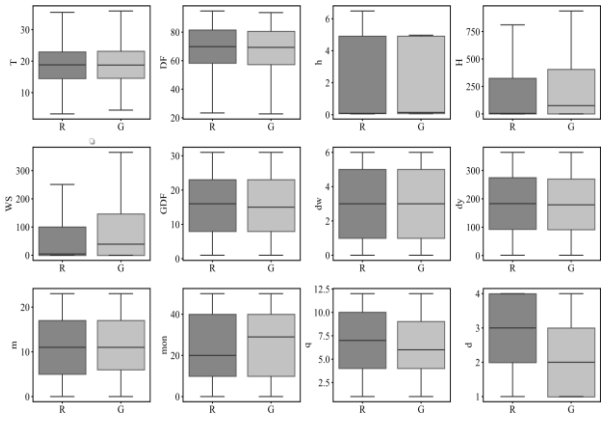

Fig. 3: Comparison of feature distributions between real data and generated data

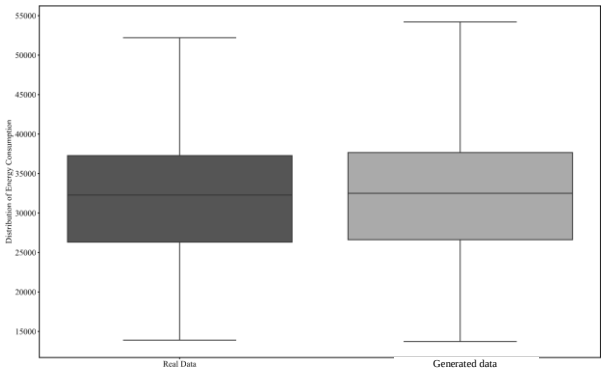

Fig. 4: Comparison of energy consumption distributions between real data and generated data

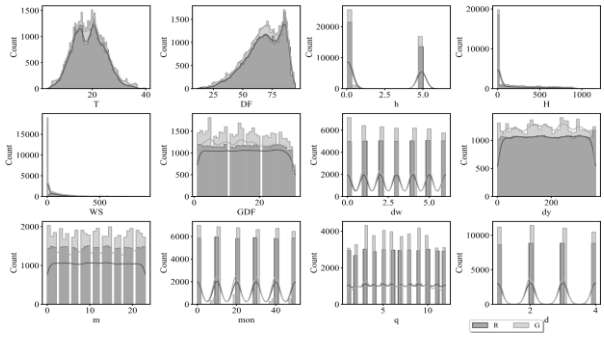

Fig. 5: Real data and generated data feature distribution comparison

For each feature, histograms and kernel density plots consistently align, indicating that the CGAN has identified correlations and patterns in the original data, generating synthetic data that closely mirrors the distribution of real data. The synthetic data is expected to be effective as a substitute for the original data in our subsequent steps, given the high degree of similarity in the feature distributions. This observation is further supported by the correlation heat map.

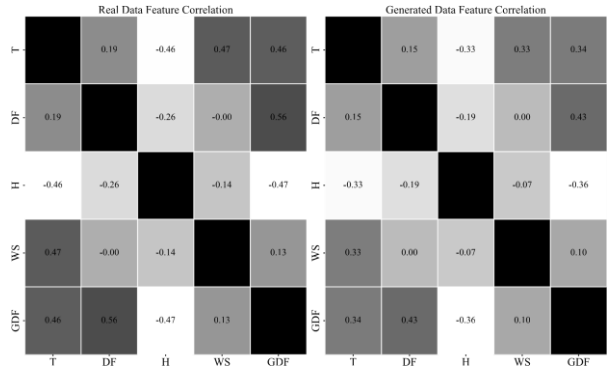

Fig. 6: Heat map correlation comparison between real data and generated data

Performance Analysis of Machine Learning Models

The comparative evaluation of each regression model's overall performance using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) on the original, generated, combined datasets and MSC-Strategy dataset is presented in this section.

Table 5 presents the results of various regression algorithms applied to data, using the RMSE metric. The results also include comparisons between the original data (R), Generated data (G), Combined data (C), and MSC-strategy data (MSC). Additionally, the last column in each table shows the percentage difference between the RMSE of the original data and the MSC data.

Based on the RMSE analysis results for each algorithm, the summarized findings are as follows: Our MSC strategy outperforms the original data by 2.47% for DT. It is slightly better than the original data for GB with an improvement of 0.97% and a noticeable improvement of 3.95% for KNN. We also show an improvement in prediction for RF, SVR, and XGB with 0.95, 0.37 and 0.29%, respectively. These results highlight the effectiveness of our strategy with all models, proving to be a valuable approach for enhancing model performance compared with other training data.

The Mean Absolute Error (MAE) values for various regression algorithms are displayed in Table 6. The MAE values for the following are given: MAER for the original data, MAEG for generated data, MAEC for combined data, and MAEMSC for a MSC-Strategy. Furthermore, the MAER vs. MSC (%) percentage difference between MAER and MAEMSC is given.

The value of MAE can be analyzed to obtain important insights into how different regression algorithms are affected by an MSC approach. Across all models, DT and KNN show better performance when using the MSC approach. GB, SVR, XGB, and RF on the other hand, exhibit negligible variations in MAE, suggesting that the MSC method has less of an impact on its performance.

Figures 7-8 display information about the number of splits for different regression algorithms in terms of RMSE and MAE given by the MSC Strategy. Also, Table 7 presents the optimal number of splits. The results show that the number of splits varies across regressors. Generally, more splits result in lower RMSE and MAE, indicating better model performance. DT and GB perform best with 7 splits. KNN and RF achieve the lowest RMSE and MAE with 3 and 8 splits, respectively. SVR and XGB have the lowest MAE with 9 splits.

Table 5: Comparing RMSE of original, generated, combined data, and MSC data

Algorithm	RMSER	RMSEG	RMSEC	RMSESC	RMSER vs. MSC (%)
DT	2686.51	2940.00	2813.47	2620.16	2.47
GB	1945.23	2272.87	2080.21	1926.34	0.97
KNN	2135.82	2157.80	2234.92	2051.53	3.95
RF	1918.45	2188.77	2108.52	1900.17	0.95
SVR	3333.64	3380.00	3335.56	3321.15	0.37
XGB	846.31	1411.86	1191.76	843.85	0.29

Table 6: Comparing MAE of original, generated, combined data and MSC data

Algorithm	MAER	MAEG	MAEC	MAEMSC	MAER vs. MSC (%)
DT	2008.36	2223.55	2129.95	1967.74	1.51
GB	1425.00	1692.38	1544.42	1414.22	0.55
KNN	1506.26	1524.62	1526.57	1421.70	3.96
RF	1383.64	1644.00	1565.61	1373.16	0.55
SVR	2609.30	2658.00	2610.69	2601.38	0.24
XGB	595.90	1001.44	857.50	591.56	0.51

Table 7: Summary of the optimal number of splits for regressors

Regressor	RMSE	MAE
DT	7	7
GB	7	7
KNN	3	3
RF	8	8
SVR	2	9
XGB	6	9

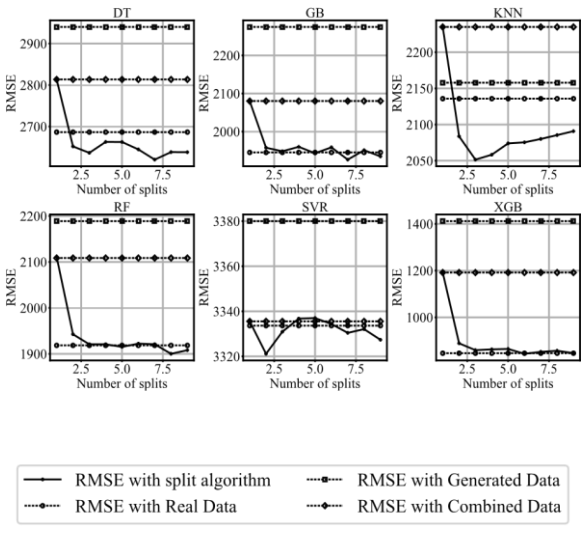

Fig. 7: RMSE value vs number of splits

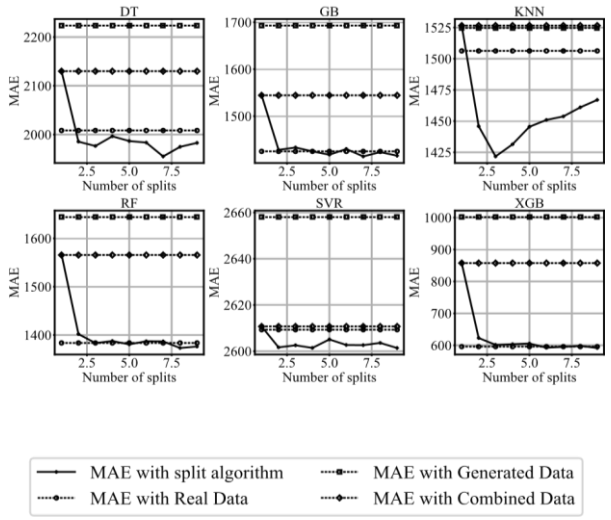

Fig. 8: MAE value vs number of splits

Conclusion

In conclusion, our study conducted a comparative evaluation of regression models on original, generated, and combined datasets and the MSC-strategy dataset, shedding light on the benefits and challenges of using CGAN-generated data for regression applications. The results presented compelling insights into the effectiveness of CGAN in addressing data limitations and MSC-Strategy data in enhancing model performance. Our analysis revealed that the RMSE and MAE were minimized with the MSC-strategy dataset, which includes both the original data and a sub-part of CGAN-generated data. The MSC-strategy dataset demonstrated superior performance compared to the models trained on the original dataset alone and the models trained on CGAN-generated data separately.

Our finding emphasizes the potential advantages of using CGAN as a means of data augmentation to address data limitations, especially in scenarios where obtaining a large, diverse, and representative dataset is challenging or expensive. The integration of our MSC strategy with CGAN-generated data contributed valuable information that complemented the original data, leading to improved model accuracy and better predictions. The MSC-strategy data keeps the models capturing the underlying data distribution more effectively, enhancing their generalization capabilities.

However, we acknowledge that the use of CGAN-generated data is not without challenges. Careful validation and quality control are essential to ensure that the generated data accurately represents the original data distribution and does not introduce biases or artifacts. Additionally, the choice of CGAN architecture and hyperparameters can significantly impact the quality of the generated data and the accuracy of our MSC strategy.

Overall, our study highlights the potential of MSC-strategy combined with CGAN-generated data in enhancing regression model performance and indicates that incorporating such data into the modeling process can be a valuable strategy for various regression tasks. It opens an avenue for further research and experimentation to optimize the number of splits of the strategy and explore the applicability of combined datasets in different domains.

In our future work, examining how different configurations affect data generation quality could offer valuable insights into optimizing the performance of CGANs for specific tasks, potentially leading to further improvements in prediction accuracy. Also, we tested our strategy with another set of generated data.

Acknowledgment

Thank you to the publisher for their support in the publication of this research article. We are grateful for the resources and platform provided by the publisher, which have enabled us to share our findings with a wider audience. We appreciate the efforts of the editorial team in reviewing and editing our work and we are thankful for the opportunity to contribute to the field of research through this publication.

Funding Information

The authors have not received any financial support or funding to report.

Author's Contributions

Abdelfattah Abassi: Conceptualization, conducted all the experiments and investigated the data, analysis, and written original drafted preparation.

Brahim Bakkas: Organized the study reviewed the results and checked the experiments done, validated and funded acquisition.

Mostapha El Jai: Reviewed and revised the data analysis and contributed to written the manuscript.

Ahmed Arid and Hussain Benazza: Supervision, reviewed and funded acquisition.

Ethics

The authors confirm that this article has not been published in any other journal. The corresponding author confirms that all the authors have read and approved the manuscript. Additionally, no ethical issues are involved in the manuscript or the dataset, and no conflicts of interest are involved.

References

Alloza, C., Knox, B., Raad, H., Aguilà, M., Coakley, C., Mohrova, Z., ... & Batech, M. (2023). A Case for Synthetic Data in Regulatory Decision-making in Europe. *Clinical Pharmacology and Therapeutics*, *114*(4), 795-801. https://doi.org/10.1002/cpt.3001

Alqahtani, H., Kavakli-Thorne, M., & Kumar, G. (2021). Applications of generative adversarial networks (gans): An updated review. *Archives of Computational Methods in Engineering*, *28*, 525-552. https://doi.org/10.1007/s11831-019-09388-y

Ashrapov, I. (2020). Tabular GANs for uneven distribution. *arXiv Preprint arXiv:2010.00638*. https://doi.org/10.48550/arXiv.2010.00638

Calderaro, A. (2015). Book review: Big data: A revolution that will transform how we live, work and think. https://doi.org/10.1177/0163443715596318b

Chatterjee, S., & Byun, Y. C. (2023). A synthetic data generation technique for enhancement of prediction accuracy of electric vehicles demand. *Sensors*, *23*(2), 594. https://doi.org/10.3390/s23020594

El Emam, K., Mosquera, L., & Hoptroff, R. (2020). *Practical synthetic data generation: Balancing privacy and the broad availability of data*. O'Reilly Media. ISBN-10: 1492072745.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139-144. https://doi.org/10.1145/3422622

Hernandez, M., Epelde, G., Alberdi, A., Cilla, R., & Rankin, D. (2022). Synthetic data generation for tabular health records: A systematic review. *Neurocomputing*, *493*, 28-45. https://doi.org/10.1016/j.neucom.2022.04.053

Kenney, J. F. (1939). Mathematics of statistics part one. D.van Nostrand Company Inc Toronto New York. https://archive.org/details/dli.ernet.5933

Krippendorff, K. (2009). Testing the reliability of content analysis data. *The Content Analysis Reader*, 350-357.

Ladeira Marques, M., Moraes Villela, S., & Hasenclever Borges, C. C. (2020). Large margin classifiers to generate synthetic data for imbalanced datasets. *Applied Intelligence*, *50*(11), 3678-3694. https://doi.org/10.1007/s10489-020-01719-y

Lanovaz, M. J., & Hranchuk, K. (2021). Machine learning to analyze single-case graphs: A comparison to visual inspection. *Journal of Applied Behavior Analysis*, *54*(4), 1541-1552. https://doi.org/10.1002/jaba.863

Lu, Y., Chen, D., Olaniyi, E., & Huang, Y. (2022). Generative adversarial networks (GANs) for image augmentation in agriculture: A systematic review. *Computers and Electronics in Agriculture*, *200*, 107208. https://doi.org/10.1016/j.compag.2022.107208

McKinney, W. (2015). Pandas, python data analysis library. http://pandas.pydata.org, 3-15.

Pan, Z., Yu, W., Yi, X., Khan, A., Yuan, F., & Zheng, Y. (2019). Recent progress on generative adversarial networks (GANs): A survey. *IEEE Access*, *7*, 36322-36333. https://doi.org/10.1109/ACCESS.2019.2905015

Rajotte, J. F., Bergen, R., Buckeridge, D. L., El Emam, K., Ng, R., & Strome, E. (2022). Synthetic data as an enabler for machine learning applications in medicine. *Iscience*, *25*(11). https://doi.org/10.1016/j.isci.2022.105331

Salam, A., & El Hibaoui, A. (2018, December). Comparison of machine learning algorithms for the power consumption prediction: Case study of tetouan city. *In 2018 6th International Renewable and Sustainable Energy Conference (IRSEC)*, (pp. 1-5). IEEE. https://doi.org/10.1109/IRSEC.2018.8703007

Saxena, D., & Cao, J. (2021). Generative adversarial networks (GANs) challenges, solutions and future directions. *ACM Computing Surveys (CSUR)*, *54*(3), 1-42. https://doi.org/10.1145/3446374

Soori, M., Arezoo, B., & Dastres, R. (2023). Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cognitive Robotics*. https://doi.org/10.1016/j.cogr.2023.04.001

Sorin, V., Barash, Y., Konen, E., & Klang, E. (2020). Creating artificial images for radiology applications using generative adversarial networks (GANs) a systematic review. *Academic Radiology*, *27*(8), 1175-1185. https://doi.org/10.1016/j.acra.2019.12.024