SAGAD: Synthetic Data Generator for Tabular Datasets

Henrique Matheus F. da Silva, Rafael S. Pereira, Fabio Porto

[1]DEXL-Laboratório Nacional de Computação Científica (LNCC)
25651-075, Petrópolis - RJ - Brasil

{Matheusf,Rpereira,fporto}@lncc.br

Abstract. *The accuracy of machine learning models implementing classification tasks is strongly dependent on the quality of the training dataset. This is a challenge for domains where data is not abundant, such as personalized medicine, or unbalance, as in the case of images of plant species, where some species have very few samples while others offer large number of samples. In both scenarios, the resulting models tend to offer poor performance. In this paper we present two techniques to face this challenge. Firstly, we present a data augmentation method called SAGAD, based on conditional entropy. SAGAD can balance minority classes in conjunction with the increase of the overall size of the training set. In our experiments, the application of SAGAD in small data problems with different machine learning algorithms yielded significant improvement in performance. We additionally present an extension of SAGAD for iterative learning algorithms, called DABEL, which generates new samples for each epoch using an optimization approach that continuously improves the model's performance. The adoption of SAGAD and DABEL consistently extends the training dataset towards improved target classification performance.*

1. Introduction

Learning algorithms generalize the information captured from a large number of samples enabling the construction of models for challenging scenarios, such as: patient diagnosis; plant species classification; and drug recommendation, just to name a few. Despite its adoption in a number of fields, there is still a group of applications where the amount of available data is not enough. As an example, consider athlete's training suggestions [Porto et al. 2012], which is very dependent on the characteristics of each individual and similar to what is found in personalized medicine. In those few samples scenarios, learning algorithms are not given enough samples so that the model achieve desirable generalization capability.

Problems associated with small number of instances in the training set have been previously studied ([Vanegas et al. 2018], [Prince and De Vos 2018]). Also, similar problems have been investigated in the context of sample unbalance [Cugliari et al. 2019].

To tackle these issues, we propose SAGAD a method based on data augmentation to mitigate the learning algorithm generalization problems caused by unbalanced and small data. The SAGAD method preserves the relationships between attribute values, as well as their distributions, when applying data augmentation. In addition, for machine learning problems that use iteractive algorithms, we present DABEL (Data Generation Based on Complexity per Classes) to optimize the data generated quality towards model performance improvement.

Previous work in machine learning that use data augmentation based techniques, focus on generating data to feed the algorithm based on some heuristics, independent of the target learning algorithm [Shorten and Khoshgoftaar 2019]. When only considering SAGAD, our work becomes similar to the usual strategies, as the adopted heuristic is based on feature distribution and relationship between features. However, as we consider DABEL, we additionally include an optimization step that guides the data augmentation process towards improving the performance of the existing model.

The remainder of this paper is organized as follows. Section 2 presents some preliminary concepts. Next, Section 3, presents the SAGAD and DABEL algorithms. Then, in Section 4, we describe the experimental set-up and, in section 5, we discuss the evaluation results. Finally, the conclusions and future work are presented in section 6.

2. Preliminaries

2.1. Machine Learning

Machine learning is a subfield of artificial intelligence, which aims to create models based on seen data. Two large subgroups of machine learning are known as supervised learning and unsupervised learning. The first aims to learn a function $y = F(x)$ using a set of labeled data, (x_i, y_i), during a process known as training. At each sample i, x_i describes a list of features and y_i is the target variable. The learned function $F(x)$, once trained, can be executed on new values of x to predict what should be its corresponding y value. Some examples of learning algorithms include: Logistic Regression; Decision Trees; Support Vector Machines; Random Forests; and Neural networks, among others. Unsupervised learning, on the other hand, considers unlabeled data. In this scenario, one aims to apply an algorithm on the available data in order to find some form of structure or identify groups. Some examples of unsupervised learning algorithms include: K-means/K-modes; Auto-encoders; DBSCAN, among others.

2.2. Data Argumentation

Data augmentation is an area of study consisting of techniques to expand the sample set. The technique introduces new samples that did not originally exist in the data, ensuring that statistical similarity occurs between the samples. The approach is useful for various methods and domains [Van Dyk and Meng 2001]. Data argumentation techniques are commonly used in machine learning, in scenarios where available data do not cover the different patterns occurring on a domain. Its use can be employed to data of several modalities.

In this context, tabular data is frequent in several domains. For example, personalized medicine-based health treatment models [Zhang et al. 2019], [Chen et al. 2017] are created for each individual patient, and since collecting many samples from a individual is not feasible, this leads to a small data problem. Thus, data augmentation techniques can become useful to provide more data to create generalizable models in small data domains.

This is, for a simple dataset D (composed by two attributes and two samples) this initial data augmentation strategy is based on the generation of a new sample (denominated d_3) as the mean of previous values. The synthesis of data by the mean procedure unfortunately has several problems. One of them is that it ignores the correlation among attributes, analyzing the attributes individually. When analyzing the attributes individually, linearity between them is assumed. In other words, there is a risk that the new samples would not belong to the original distribution of the data.In this sense, a new set of techniques was proposed to create synthetic data correlated with their attributes.

In this direction, it is worth to mention the growth of popular tools like SMOTE [Chawla et al. 2002] that proposes to generate data by correlating its attributes. SMOTE finds points of the same class and interpolates among them, to generate new samples. Initially, SMOTE was proposed only for continuous data and later updated to support categorical data [Mukherjee and Khushi 2021].

As this particular example, several DA techniques attempt to incorporate statistical techniques for the construction of new samples with the premise of obtain better results, for example GANs for tabular data (GANT) [Ashrapov 2020]. Originally, GANs was developed for imaging, but recently received modifications to the other domains. Table 1 presents a set of DA techniques.

Data Argumentation in tabular data		
Techniques	Categorical data	Machine learning
SAGAD	X	
SMOTE		
FAST-DAD	X	X
Mean using line space		
SMOTE-ENC	X	
GAN TABULAR	X	X
DLTD	X	X

Table 1. Some of most common Data Augmentation techniques. In the table is marked when techniques are able to be applied on categorical data and which are based on machine learning techniques.

Table 1 classifies the different data augmentation techniques as: support for tabular data, and whether the approach adopts any ML component. The use of ML algorithms requires some prior knowledge. Joining these principles with statistical techniques increases the complexity of the method.

In this scenario, we propose a new technique in data augmentation for categorical data, called SAGAD. This strategy is based on principles of probability with more flexible concepts than techniques using ML.

SMOTE has as its primary hypothesis that a linear interpolation between two neighbors of the same class generates only samples of the same class. This Hypothesis is not assumed in SAGAD. We would argue that this hypothesis may not hold when using points that are near a border between classes in the feature space. SMOTE also does not have a synthetic sample validation step where one can stipulate a minimum threshold of confidence that the new sample actually belongs to the desired class.

3. Methodology

In this section, we introduce the SAGAD data augmentation method for tabular data. It consists of using the relationships among the features of the training dataset to condition the generation of new samples. Additionally, we introduce the DABEL (Data generAtion Based on complExity per cLasses) method. DABEL is applied on neural networks and aims to optimize the data generated by considering how ambiguous different classes are.

3.1. SAGAD

Consider a dataset D, with $D = (a_1, a_2, \ldots, a_n)$, such as a_u, $1 \leq u \leq n$, compose D schema attributes. Also consider that $a_u \subset Dom_z$, where Dom_z is a set of values, called domain values. Now, consider additionally that D is used to train a learning algorithm A, for example: decision tree, Support Vector Machine (SVM), etc., producing a machine learning-based model M. Moreover, $card(D)$ indicates the number of samples r in D, and $card(a_u)$ the number of distinct values contained in the a_u.

The problem studied is to produce a dataset $D' = (a_1, a_2, \ldots, a_n)$, with $schema(D) = schema(D')$ and $card(D') >> card(D)$. Consider P a probability distribution function (PDF). We want to obtain D' such that in the process of extending D, the probability distribution of values in attributes D is kept, that is $P(a_u) = P(a'_u)$ for all $1 \leq u \leq n$. Another property observed is $P(a_u|a_z) = P(a'_u|a'_z)$, $u \neq z$, so that the relationships between the (a_u, a_z) pairs are maintained. To model the pdfs $P(a_u)$ and $P(a_u|a_z)$, we use the histogram technique to approximate the P function by a set of rectangles. To determine the optimal number of rectangles needed to represent P, the Sturges[Sturges 1926] method was used. The process is depicted in algorithm 1, in which we present the process of creating one sample. To create b samples of a given class, one needs to execute the algorithm b times.

In algorithm 1, D is the dataset used in the data synthesis process. The $Target\_Class\_value$ c is the class value that novel samples should belong to, and $Target\_attribute$ g is the attribute name corresponding to the classification target class in the dataset. In line two, α receives the subset of D with rows conforming to the target class "c", target attribute $g = c$. In the first loop, the algorithm traverses the set of attributes of the dataset. $hist$ is a function that generates the histogram of an attribute a_i. It returns a vector that denotes the start of each histogram cell, H, as well as the amount of values contained in the corresponding cell, k. In lines 5-8, we normalize the k values to obtain the probability density function of the column. We then calculate from it the cumulative density function CDF.

We use a random number generator function $rand()$ to generate a value in the $[0, 1]$ interval, using a uniform distribution. Given that the CDF ranges from the same interval, we can then use the random value to map to the CDF interval. The new $sample$ value, line 17, is randomly picked in the interval$[H_{index}$,$H_{index+1}]$. The values of the interval are then stored into the vectors ψ and ϕ in order to validate the sample at line 22.

Once a sample value has been generated, in line 20 we filter α to only keep the rows contained in the interval $[H_{inter}, H_{inter+1}]$. The process repeats until all features have a value, and the set of values generated constitute a novel sample. After the synthetic sample is generated, we test the choices that led to the novel sample against the dataset containing all classes, in order to find what is the probability the sample belongs to the desired class. A parameter $prob$ is provided to the function in order to reject samples that do not have at least $prob$ probability of being of the desired class.

Algorithm 1 SAGAD algorithm

```
$SAGAD(\ Dataset\ D, Target\_Class\_value\ c, Target\_attribute\ g, cutoff\ prob)$
$\alpha \leftarrow \{x \in D |\ x.g = c\}$
for $i = 1, \cdots, n$ do
    $(H, k) \leftarrow hist(a_i)$
    $Sum \leftarrow \sum k$
    for $j = 1, \cdots, m$ do
        $k_j \leftarrow \frac{k_j}{sum}$
    end for
    $CDF \leftarrow cumsum(k)$
    $\delta \leftarrow rand(U[0, 1])$
    $index \leftarrow 1$
    for $j = 2, \cdots, m$ do
        if $CDF_j \leq \delta$ then
            $index \leftarrow j$
        end if
    end for
    $sample_i \leftarrow rand(U[H_{index}, H_{index+1}])$
    $\phi_i \leftarrow H_{index}$
    $\psi_i \leftarrow H_{index+1}$
    $\alpha \leftarrow \{x \in \alpha\ |H_{index} \leq x.ai \leq H_{index+1}\}$
end for
$sample \leftarrow Validation\ synthetic\ sample(D, sample, \phi_i, \psi_i, prob)$
Return sample
```

3.2. Optimizing generated candidates using DABEL

In the context of a multi-class classification problem, an important result obtained from the data augmentation procedure is to separate instances whose mappings to classes are ambiguous. In such a context, by expanding the number of instances in ambiguous regions of the data, we expect to improve the classification model performance.

An example of ambiguity can be seen in Figure 1, whereas we present a dataset with 3 classes, where one is very isolated from the others and requires very few samples to be classified. On the other hand, the green and blue points are very close to each other, which requires a larger number of samples to better separate them.

Given that the ambiguity among classes are hard to foresee, we believe that data augmentation optimization is a process that should be considered along model optimization. We choose to develop DABEL to extend SAGAD, when considering iterative algorithms. The algorithm relies on the iterative computation of the loss function per epoch.

DABEL is combined with SAGAD so that new samples are generated for each epoch, in the hope of guiding data augmentation towards prediction accuracy improvement. During this process, we favor more ambiguous classes in the data generation process. The number of generated samples for a class t is presented in equation 1.

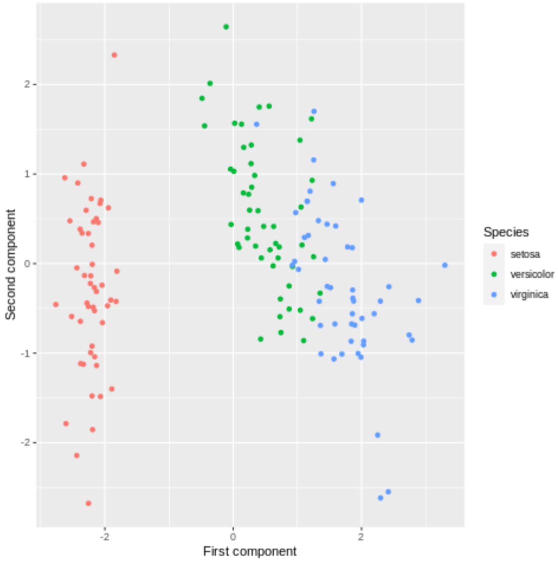

Figure 1. PCA visualization in iris dataset

$$Sample_c = S * \frac{\mathrm{loss_c}}{\sum_{L=1}^{n} \mathrm{loss_L}} \tag{1}$$

where

- $Sample_c$ = Synthetic data for Class c, generated for next epoch;
- S= Total number of samples generated for all classes;
- $Loss_c$ = Loss for class c;
- $Loss_L$ = Loss for class L.

By adopting our proposed method for generating the training set, we aim to feed more samples of harder classes to the network until it can achieve a satisfiable loss value on the validation set for all classes. We expect that by using such an approach the network can focus on harder classes that present greater ambiguity during the training process.

In Figure 2, we present the pipeline which implements SAGAD along with DABEL, expanding a dataset to feed it into a neural network. The dataset is divided into training, validation, and testing.

The training data is the input to SAGAD for data generation. The output from SAGAD is then used for training the neural network. Later the model is validated with the validation data. At the end of the first epoch, the loss is calculated over each class using the validation dataset. At this point, we use the loss as a measure of class ambiguity. DABEL uses it to inform SAGAD which classes should be prioritized for the data generation process for the next epoch. The process repeats until the model converges by reaching a validation loss smaller than some threshold the user considers or the maximum number of epochs are executed. Then the model can be executed in the test data to evaluate its generalization capability. Finally, observe that the model used during the DABEL step is a preliminary one. Once the data have been augmented, the training process can be resumed by training the learner in the augmented training data.

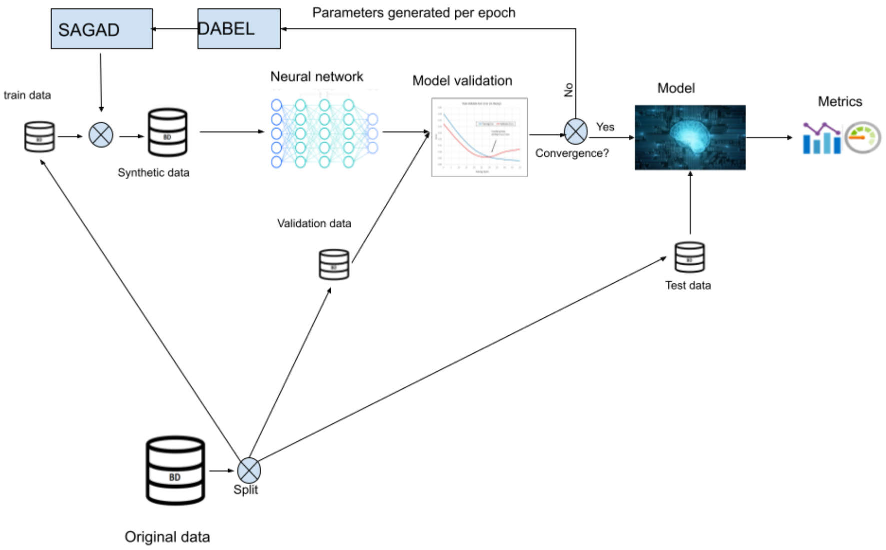

Figure 2. Pipeline about data generation process using SAGAD and DABEL

4. Experiments

In this section, we present the datasets used in the validation of this work, as well as the methodology adopted to simulate the scenario of small data. Lastly, we present the machine learning pipeline designed to perform the experiments.

4.1. Experimental Set-up

All experiments were run using google collaboratory and deep learning models were trained using a GPU NVIDIA Tesla K80 by using the keras package. When training machine learning models, we used the R caret package [from Jed Wing et al. 2018]. Our data augmentation method named SAGAD was also implemented in R and is published under the name AugmenterR [S. Pereira et al. 2021] in CRAN. All caret models were optimized using cross-validation with 5 folds.

In this work, the datasets Iris[Dua and Graff 2017a] and Wine Quality [Dua and Graff 2017b] were used, with 150 and 6,497 examples, respectively. Iris and Wine are benchmark datasets balanced an unbalanced (69% for white class and 31% for red class), respectively. Initially, we sample the datasets, in order to simulate the problem of small data. For both datasets, the samples present sizes $N = 15$, $N = 25$ and $N = 45$, respecting the original imbalance of classes.

When training the model, the datasets were first split into train and test. The train set contained N samples and the test set contained the rest of the data. The training set was then further split into 70% and 30% for training and validation. The model then will be optimized on the training set and evaluated on the test set.

In oder to create novel samples we use the following process: firstly, 1.500 data points are generated using the training dataset in all machine learning algorithms, for SMOTE, GANT and SAGAD. Secondly, in the method known as DABEL described in section 3.2, 1500 datapoints were generated per epoch, where the number of samples per class was calculated according to equation 1.

The parameters used for data creation used the default settings mentioned in the documentation of each of the DA techniques, except in the training process for GANT, in which the number of epochs was reduced from 500 to 20, to simulate the same computational cost of SAGAD. We repeat the process 10 times and evaluate it using f1-score.

To verify if there is statistical difference between means on the analysed techniques, we performed the *One Tailed T* statistical test. We considered a *p-value* of 0.05 to reject a NULL hypothesis that SAGAD mean metric value would be smaller or equal to the other methods. We summarise those results in Tables where ”*” represents the alternative hypothesis, while ”-” represents the NULL hypothesis.

5. Results

Figure 3 depict the comparison of Wine dataset for 15, 25 and 45 samples. In Figure 4, we present the same result for the Iris dataset.

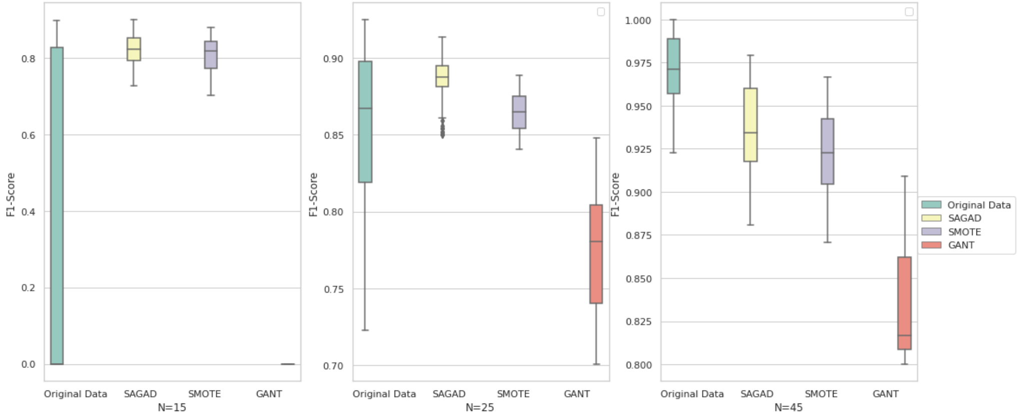

Figure 3. F1-Score in Wine dataset for decision tree

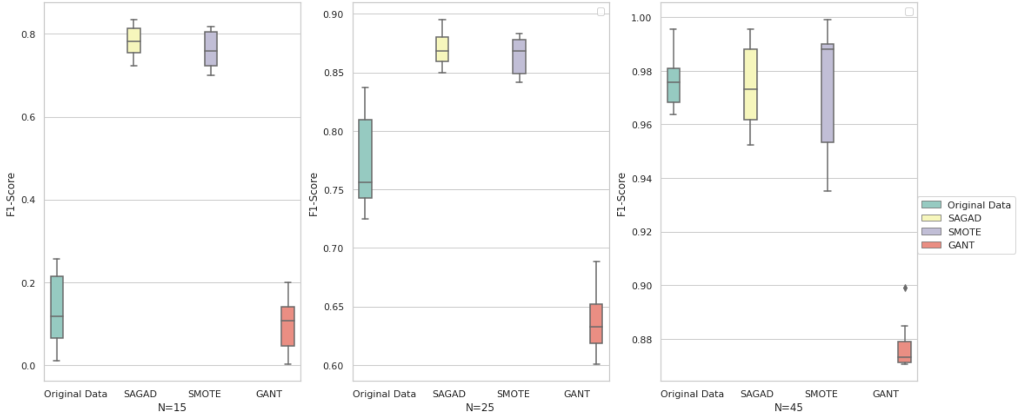

Figure 4. F1-Score in Iris dataset for decision tree

Analyzing the bloxplots of the metrics in both cases, the smaller the number of samples in the dataset, the better the performance of SAGAD. In all figures for $N = 15$ metrics computed on the original datasets achieved results greater than 50% prediction, something different in the adoption of SAGAD and SMOTE. In cases $N = 25$, both metric results show higher values when SAGAD and SMOTE is applied. For 45 samples, the trained model using SAGAD, SMOTE and the original dataset show equivalent performance.

In Tables 3 and 2 we can visualize the *One Tailed T* statistical test results.

N	SAGAD x Original	SAGAD x SMOTE	SAGAD x GANT
15	*	-	*
25	*	-	*
45	-	-	*

Table 2. Comparison of p-value for Iris dataset for decision tree

N	SAGAD x Original	SAGAD x SMOTE	SAGAD x GANT
15	*	-	*
25	*	-	*
45	-	-	*

Table 3. Comparison of p-value for Wine dataset for decision tree

Note that the implementation of SAGAD obtained a better result for $n = 15$ and $n = 25$ than original data in both tables. In all cases, SAGAD obtained a statistical difference was in comparison to GANT and not for SMOTE.

Figure 5 depict the comparison of Wine dataset for 15, 25 and 45 samples. In Figure 6, we present the same result for the Iris dataset.

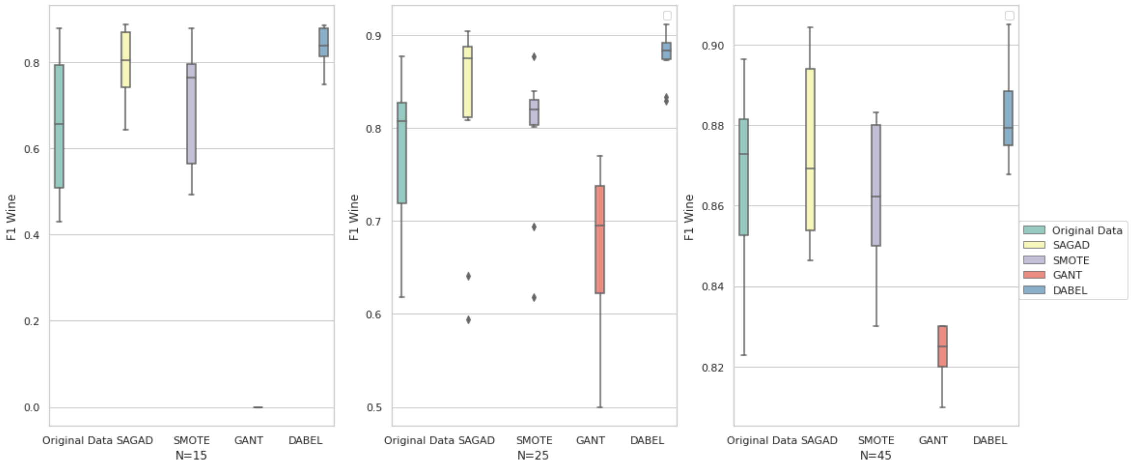

Figure 5. F1-Score in Wine dataset for neural networks

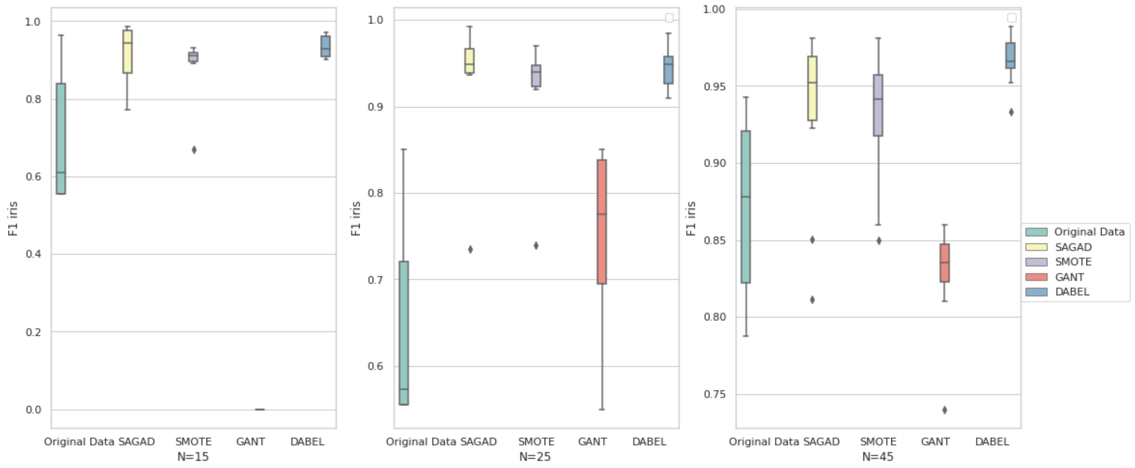

Figure 6. F1-Score in Iris dataset for neural networks

Analyzing the bloxplots of the metrics in both cases, SAGAD, DABEL and SMOTE proved to be effective for solving the proposed problem when data is small compared to the original data and GANT. We highlight that in none of the scenarios, the quality of the model worsens when using the proposed methodology. We note that as N gets smaller SAGAD,SMOTE and DABEL performance outshines the other methods.

In Tables 4 and 5 we can visualize the *One Tailed T* statistical test results.

N	DABEL x Original	DABEL x SAGAD	DABEL x SMOTE	DABEL x GANT
15	*	-	-	*
25	*	-	-	*
45	*	-	-	*

Table 4. Comparison of p-value for Wine dataset

N	DABEL x Original	DABEL x SAGAD	DABEL x SMOTE	DABEL x GANT
15	*	-	-	*
25	*	-	-	*
45	-	-	-	*

Table 5. Comparison of p-value for Iris dataset

In the tests with Wine, DABEL obtained statistical difference in relation to original data and GANT, in all cases. However, in Iris the only case that DABEL did not obtain a statistical difference in comparison to wine is $N = 45$ for both metrics. To verify if there is a difference between variances comparing SAGAD and DABEL we performed the *one tailed F statistical test* assuming a null hypothesis that DABEL variance is smaller than SAGAD variance. Results are presented on table 6. We also test for standard deviation for SAGAD vs DABEL but due to lack of space we do not present the table results.

Results comparing variance for DABEL and SAGAD		
N	Iris	Wine
15	*	*
25	*	*
45	-	*

Table 6. Comparison between variances between DABEL and SAGAD

By visualizing Table 6, we can notice that with the exception of case N=45 for Iris, we coud not reject the null hypothesis that DABEL variance is smaller than SAGAD variance.

The tables 7 and 8 present the same tests performed in the decision tree with the Logistic regression, Random Forest and SVM for accuracy and f1-score metrics.

Methods	$N = 15$	$N = 25$	$N = 45$
Logistic Regression	*	-	-
SVM	*	*	-
Random Forrest	*	-	-

Table 7. Test SAGAD vs original data in machine learning algorithms

Methods	$N = 15$	$N = 25$	$N = 45$
Logistic Regression	*	-	-
SVM	*	*	-
Random Forrest	*	-	-

Table 8. Test SAGAD vs original data in machine learning algorithms

We emphasize that these are preliminary results given time constraints, and future research comparing these methods on a larger set of datasets is advisable.

6. Conclusion

In this paper, we present the SAGAD algorithm for data augmentation applied to tabular data. It is already available on CRAN for the R language with more than 2000 downloads to date. We presented tests using the Iris and Wine datasets. We also present DABEL, SAGAD extension for iterative learning algorithms. By using SAGAD and DABEL we were able to improve results in small data scenarios where models failed to generalize. Future work should test these methods for more complex datasets, as well as compare it against other data augmentation techniques. We also consider expanding SAGAD for other data modalities.

References

Ashrapov, I. (2020). Gans for tabular data. https://github.com/Diyago/GAN-for-tabular-data.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Chen, M., Hao, Y., Hwang, K., Wang, L., and Wang, L. (2017). Disease prediction by machine learning over big data from healthcare communities. *IEEE Access*, 5:8869–8879.

Cugliari, G., Benevenuta, S., Guarrera, S., Sacerdote, C., Panico, S., Krogh, V., Tumino, R., Vineis, P., Fariselli, P., and Matullo, G. (2019). Improving the prediction of cardiovascular risk with machine-learning and dna methylation data. In *2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–4.

Dua, D. and Graff, C. (2017a). UCI machine learning repository.

Dua, D. and Graff, C. (2017b). UCI machine learning repository.

from Jed Wing, M. K. C., Weston, S., Williams, A., Keefer, C., Engelhardt, A., Cooper, T., Mayer, Z., Kenkel, B., the R Core Team, Benesty, M., Lescarbeau, R., Ziem, A., Scrucca, L., Tang, Y., Candan, C., and Hunt., T. (2018). *caret: Classification and Regression Training*. R package version 6.0-80.

Mukherjee, M. and Khushi, M. (2021). Smote-enc: A novel smote-based method to generate synthetic data for nominal and continuous features. *Applied System Innovation*, 4(1):18.

Porto, F., de Carvalho Moura, A. M., da Silva, F. C., Bassini, A., Palazzi, D. C., Poltosi, M., de Castro, L. E. V., and Cameron, L. C. (2012). A metaphoric trajectory data warehouse for olympic athlete follow-up. *Concurr. Comput. Pract. Exp.*, 24(13):1497–1512.

Prince, J. and De Vos, M. (2018). A deep learning framework for the remote detection of parkinson's disease using smart-phone sensor data. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3144–3147. IEEE.

S. Pereira, R., ferreira da silva, H. M., and A.M Porto, F. (2021). *AugmenterR: Data Augmentation for Machine Learning on Tabular Data*. R package version 0.1.0.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48.

Sturges, H. A. (1926). The choice of a class interval. *Journal of the American Statistical Association*, 21(153):65–66.

Van Dyk, D. A. and Meng, X.-L. (2001). The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50.

Vanegas, M. I., Ghilardi, M. F., Kelly, S. P., and Blangero, A. (2018). Machine learning for eeg-based biomarkers in parkinson's disease. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2661–2665.

Zhang, S., Bamakan, S. M. H., Qu, Q., and Li, S. (2019). Learning for personalized medicine: A comprehensive review from a deep learning perspective. *IEEE Reviews in Biomedical Engineering*, 12:194–208.