Generating synthetic 2019-nCoV samples with WGAN to increase the precision of an Ensemble Classifier

Arnon Bruno Ventrilho dos Santos, Deborah Ribeiro Carvalho
Pontifícia Universidade Católica do Paraná (PUC-PR)
E-mails: asantos.quantum@gmail.com, drdrcarvalho@gmail.com

Abstract: The objective of this research is to present an alternative data augmentation technique based on WGAN to improve the precision in detection of positive 2019-nCoV samples, as well as compare it with other traditional data augmentation techniques, using a dataset composed of results of exams from individuals tested for COVID-19 in a hospital in Brazil. Given the data imbalance, we presented a gradient-boosted decision tree (GBDT) classifier with the preprocessed data in 13 different oversampling training scenarios, using SMOTE, ADASYN, Random Over Sampling, and WGAN to augment positive samples, as well as no augmentation at all. All over-sampling scenarios are set so that a mixture of real and synthetic samples is presented to the classifier. GBDT classifier was then trained in all scenarios with stratified k-fold (k=10), and its hyperparameters were optimized for the highest possible f1-measure with the random search algorithm for 20000 epochs. A hold-out test set was prepared by randomly removing an even number of really positive and negative samples from the training set and shuffling it for testing. GBDT classifier trained with 50% WGAN synthetic positive samples achieved a precision score of 0.967 on the test set, far outperforming all other scenarios, including similar mixture scenarios using other oversampling strategies. These results indicate that data augmentation with WGAN to oversample the minority class might be an alternative for traditional oversampling techniques, even improving a classifier's precision.

Keywords: COVID-19, Generative Adversarial Networks, Classifier.

1. INTRODUCTION

In later 2019, the world has seen the surge of a global pandemic caused by SARS-Cov-2 (also known as 2019-nCoV), a virus is known to be originated in Wuhan, China. This pandemic caused the death of dozens of thousands of people around the world until mid-2020. Methods to either treat or extinguish the disease are under development in the form of drugs or vaccines. Still, until such a solution is fully developed, people must get tested to be rapidly treated and isolated in case of infection [1]. Among the available detection mechanisms, one could use machine learning classifiers, which could point to preventive measures in advance.

Because available training samples for 2019-nCoV are not abundant, synthetically augmenting real samples emerge as a potential alternative to increase the number of training samples without depending on real individuals' infection and testing

Generative Adversarial Networks (GANs) is a Deep Learning based model in which a competitive process involving a pair of neural networks generate artificial data from random noise. Such process occurs by iteratively training a generator (G) network that presents both real samples and random noise into a discriminator (D) network, which then evaluates the quality of these samples and feed-back to G, that tries to minimize the loss by providing more realistic samples originated from random noise to D in the next interaction [2]. This process has proven to be effective in creating realistic samples in the form of images in the computer vision realm [3] and other domains such as natural language processing [4].

Some modifications had been proposed to the GAN framework since its first appearance back in 2014. Aiming to tackle the instability in the learning process of GANs and provide meaningful learning curves to ease hyperparameter adjustment, the Wasserstein GAN (WGAN) algorithm has been proposed as an improved version of the original GAN. This algorithm works

by replacing D with a Critic (C) network, applying the Wasserstein loss function and a weighted clipping to enforce a Lipschitz constraint on C [5][6]. With such improvements, the training of GANs became more stable and therefore allowed for more complex and realistic synthetic data to emerge [6].

In this research, we intend to train an ensemble classifier in different over-sampling training scenarios and measure its performance in a test set by evaluating its capacity to identify true positives concerning all positive samples using the precision score well as other secondary metrics. Our intention is also to evaluate if training a classifier with synthetic positive samples created with WGAN poses an alternative to traditional data augmentation techniques such as SMOTE [7], ADASYN [8], and Random Oversampling (ROS) [9].

2. 2019-nCoV

Known to be originated in Wuhan (China) in later 2019, 2019-nCoV is a virus that causes severe acute respiratory disease and is easily transmitted from one individual to another. Recent research [10] indicates that the reproduction number (R_0) of this virus is around 2.2, which means that each individual can infect 2.2 other individuals. To put that into perspective, the same research points that the virus responsible for the common flu (Influenza) has an estimated R_0 of 1.5. This basically suggests that the 2019-nCoV is a highly infective agent, even compatible with the deadly 1918 flu outbreak. Therefore, its early detection could help avoid the spread of such disease [10][1].

In Brazil, there were 28 different types of tests to identify the virus until mid-2020. Most of these have high sensitivity (> 95%) and high specificity (>95%). However, these tests are not widely available and may require up to many days to deliver a result due to the high demand for these tests during the pandemic [1]. In this scenario, an alternative might be to use other exams to directly or indirectly identify the disease, such as measuring creatinine results and other exams to identify anomalous combinations that might point to a 2019-nCoV infection. This type of pattern recognition is made possible by using methods that ana-lyze large amounts of data, such as machine learning algorithms [26]. However, it is important to notice that not every 2019-nCoV case is registered to further data exploration, which may indicate a lack of enough data points for statistical models and machine learning algorithms to work properly [10][11][12]. With such a perspective, augmenting positive samples with new realistic 2019-nCoV data points could be used for analysis without the need for a real infected individual.

3. DATA AUGMENTATION

Real-world datasets are usually not uniformly distributed among its target classes. This is somehow related to the noisy nature of data itself and presents challenges to machine learning classifiers, which uses samples from these data to recognize patterns and model new unseen data samples [13]. There is data augmentation among the techniques available to tackle this challenge and improve machine learning classifiers' performance [7]. This technique consists of creating new synthetic samples from real ones, which can then be used for training, thus decreasing bias by increasing the number of minority samples [13].

Data augmentation is used in domains such as computer vision when there is a lack of image data. New samples are created to increase model variance [14], or in Natural Language Processing, to increase vocabulary corpus in texts [15]. But the use of such a technique is not restricted to these domains, as it is also generally used when there is a high imbalance between classes in a collection of observations. This is generally the case in fraud detection, bioinformatics, and medical datasets [9].

Traditionally, data augmentation algorithms focus on creating new synthetic samples by averaging the distance between real data points in a space or simply creating copies of the real

samples on that same space. The first has the effect of creating completely new synthetic samples, but that might lack realism, and the second might be realistic but present no real novelty to the model [9][14]. Some of these algorithms are further explored below.

3.1 Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE [7] is a data augmentation technique that aims to oversample the minority class by finding the nearest neighbor in a group of real samples and placing a synthetic one between them. This has the effect of creating synthetic samples that are similar to the real ones. Still, by not taking the distribution of sub-classes into account, these synthetic samples might cause overlapping be-tween classes [9][13]. Although this effect might not be substantial in a low-dimensional space with only a few samples, its scalability might be severely compromised [9].

3.2 Adaptive Synthetic (ADASYN)

Adaptative synthetic sampling [8] works by measuring the difficulty in learning samples from the minority class and then creating more synthetic samples from harder to learn. This has the effect of producing new synthetic data points that are closer to the decision boundary. It also uses a nearest neighbor approach to create these new samples, so it might still be impacted by the same points mentioned earlier in SMOTE [7][9][13].

3.3 Random Oversampling (ROS)

Among the available data augmentation techniques for oversampling, ROS is the simplest. This algorithm works by making a new copy of real random samples from the minority class. This generates no real "synthetic" samples and pro-vides no novelty to a machine learning model, leading to overfitting. Still, it might be more realistic and scalable than the techniques mentioned earlier [9][16].

4. GENERATIVE ADVERSARIAL NETWORKS (GANs)

GANs [2] are a Deep Learning framework capable of learning representations of high-dimensional data and use these representations to create new unseen data based on a competitive game between two neural networks. A generator (G) is usually a deep convolutional neural network (DCNN) that feeds a discriminator DCNN (D) with random numbers extracted from a latent space, along with numbers representing the dimensions of a real sample. D is responsible for evaluating both samples (random and real) and signals back G with its findings in the form of a "real or not real" indicator. G is then stimulated to make random numbers changes from latent space to make them look more "real." G's ultimate goal is to feed D with such realistic samples from latent space that D cannot discriminate between real and random. This framework represents a zero-sum game between two "machine" players, which ultimately leads to a convergence point where both players are stimulated to improve [17]. Recent research has demonstrated that data generated using the GANs framework resulted in outstanding results [6][18], mainly in the computer vision domain.

GANs are known to be "hard" to train [5] since there is no objective way to guarantee that both G and D's loss will remain stable during the training process. This leads to "model collapse," where either G or D becomes absolutely dominant, resulting in low-quality samples emerging from this structure. This problem has been tackled with the proposal of various other GANs structure or algorithms to ease the training of GANs. One such algorithm is the WGAN model [4][5][6].

4.1 Wasserstein GAN (WGAN)

WGAN [5] aims to solve GAN training's instability by applying some slight but elegant changes to the original framework. First, it tries to minimize the Earth Mover's distance (or Wassertein-

1) by using a new D network called "Critic" (C), which is very similar to the original D. Still, instead of outputting the probability of a given sample to be real or fake, it outputs the "realness" of that sample. Secondly, it removes the log functions in both G and C's losses and finally satisfies the 1-Lipschitz constraint by clipping the weights in C. In other words, it guarantees that there are no "realness" above 1 or less than 0, which might be a potential aspect of model collapse. However, this approach is not recommended by [6], which proposes a modified version of WGAN containing gradient penalty (GP) instead of weight clipping. This model is referred to as WGAN-GP. The whole mathematical intuition behind such changes is demonstrated in [5] and [6].

5. METHODS

This research is based on an open dataset made available by the "Hospital Israel-ita Albert Einstein" (Brazil) on the online platform "Kaggle" [22]. The dataset is already anonymized, mostly standardized, and contains the exam results of patients tested for 2019-nCoV, being positive the minority class. With this research, we test different approaches to oversample the minority class of this noisy dataset, aiming to increase a classifier's capability of detecting positive cases while minimizing false positives. The data augmentation techniques presented here are SMOTE, ADASYN, ROS, representing the most conventional techniques [9] and WGAN, representing a novel approach to oversampling. All experiments were done in Python by using its data science libraries and Google's Keras for Deep Learning.

5.1 Data Preprocessing

Before generating the synthetic samples, the dataset was transformed to minimize the noise. To accomplish this, we first changed the names of all features (which ranges from patient's age to exams results) to a discretized form, where features iteratively received the name "f" followed by an integer starting in 0. After that, we removed empty-valued features. We understand that these features would not contribute to creating good quality synthetic samples. Also, some samples had only a few empty-valued features. For these samples, we opted to fill the empty values with the median of that feature on the age quantile containing the sample since exam results would look more realistic for similar ranges of age (quantiles) than if considering the overall median for all age quantiles. Features with more than 90% of empty-valued samples were also removed since we would prefer not to input the median in such a large number of samples.

Once all samples had no empty-valued features, we worked on removing features with zero-variance or features with high collinearity (>.95) on the Spearman correlation coefficient. We understand that the first presents no discrimination capabilities, while features with high collinearity are well explained by other features and could be removed to comply with parsimony. As an additional measure to rationally decrease the number of features, we applied a recursive feature elimination (RFE) algorithm that recursively combines features concerning the target and measures its impact on model output. Features that didn't achieve a minimal impact threshold can be removed without compromising the model's final performance [25].

5.2 Data Transformation and Training

We standardized its features with the database ready by removing the mean and scaling to unit variance; then, we started to create synthetic samples. For all 4 data augmentation techniques mentioned earlier, we created several synthetic samples proportional to a given percentage of real samples in a positive class. Therefore we created 30%, 50%, and 70% synthetic positive samples.

To create synthetic samples with SMOTE and ADASYN, we opted to rely on k-nearest neighbors with k=5. This would provide us with a wide range of samples, allowing for faster

calculations of Euclidean distances. In samples created with ROS, no changes have been made to the algorithm randomness, so samples created with this method rely entirely on its standard form implemented in Python's sci-kit-learn.

Samples created with WGAN were created in a totally different fashion, where a Deep Learning model needs to be trained to generate these samples. We developed the GANify Python library [24], which ease implementing the WGAN model described in [5]. Real samples and random noise are fed into C by G in batches of size 8 uniformly distributed, with a 5% probability of flipping its labels, meaning that real samples can be fed into C with the label "false," while false samples created from random noise could be fed into C with the label "real." This is reported to be beneficial for WGAN performance [5][6]. WGAN model is then trained during 1500 epochs, which was the optimal number during our experiments in terms of model convergence.

Once all synthetic samples were available, we trained a Gradient Boosted Decision Tree (GBDT) [28] classifier in 13 different combinations of real and synthetic samples for the positive class, as well as different ratios between positive and negative (Table 1), stratifying training and validation data by the target class in 10 folds, applying cost-sensitive learning (CSL) [9] on each fold. The choice for 10 folds is based on our findings suggesting that 10 might be a number where training results are stable, while the option for CSL works as an additional measure to tackle data imbalance in all scenarios. We also created a holdout test set, which is balanced and composed of the real samples that were replaced by synthetic ones on the training set and the same number of samples randomly selected from negative class, thus the different ratios on training

We then optimized the classifier hyperparameters [23] with a uniform randomized search within the possible range of values listed in Table 2 for 20000 epochs. We noticed that increasing the number of epochs resulted in small improvements in scoring performance while considerably increasing the GBDT classifier's computational time. We opted for a GBDT [28] classifier because it gives us an obvious explanation on the effect each feature had on the model result with the aid of the Shapley Additive Explanations (SHAP) model [21] as seen in Fig. 5, which greatly helps us on validating the model results. In addition to that, GBDT classifier allows different sampling algorithms such as "gradient boosted decision tree" (gbdt), "gradient-based one side sampling" (goss), and "dropout meets multiple additive regression trees" (dart), all of those provide different training strategies that might help on improving training performance [28]

Table 1. Training Scenarios.

Algorithm	% Synthetic on Positive	Train Imbalance Ratio	Test Imbalance Ratio
None	0%	1:10	1:1
ROS	30%, 50%, 70%	1:10, 1:9, 1:8	1:1
ADASYN	30%, 50%, 70%	1:10, 1:9, 1:8	1:1
SMOTE	30%, 50%, 70%	1:10, 1:9, 1:8	1:1
WGAN	30%, 50%, 70%	1:10, 1:9, 1:8	1:1

Table 2. Classifier hyperparameters and its possible values, represented in a range when numeric

Hyperparameter	Range (from - to)
n_estimators	10 - 200
min_split_gain	0 - 0.5
num_leaves	3 – 30
learning_rate	0.001 - 1
colsample_bytree	0 – 0.5

boosting_type	“gbdt”, “dart”,”goss”

5.3 Evaluation

The impact of these techniques will be assessed by comparing the classifier’s precision, recall, and f1-score on the balanced test set, after being trained with the different combinations of real and synthetic data. We opted to test on a balanced test set to have an alternative test scenario. The stratified 10-fold validation mentioned in the previous section is already validating the training in an imbalanced subset of data. The option for a balanced test set gives us an estimate of the model’s performance on a scenario that considers an accelerated spread of 2019-nCoV infection [1].

The recall score (RS) will provide us with a metric of how well the positive samples were predicted among all test samples. However, for a classifier to excel on this metric, it could simply predict all samples as positive, which would lead to a maximum recall, but with a great number of false positives (FP). To tackle this potential flaw in our evaluation method, we’re considering the precision score (PS) as our main evaluation metric. The PS provides us with a metric demonstrating the number of predicted positive samples among all real positive samples. This will give us a good idea of how well the classifier identifies true positives (TP) against FP, a critical measure for this research. As a complementary measure, we also evaluate the f1-measure, which takes the harmonic mean of both RS and PS [20]

6. RESULTS

By applying the preprocessing and transformation routines specified in the previous section, the dataset initially composed of 5644 samples and 111 features was transformed into a dataset containing the same number of samples, but with 38 features, including one target (named “f2”), which represents the final diagnostic, being “positive” for a sample tested positive for 2019-nCoV, and negative the opposite. The dataset contains 5086 negative samples and 558 positive samples (thus a 1:10 ratio). In each of those training scenarios listed in Table 1, we explored the data dispersion in a 2D scatter plot after reducing it to its 2 main components with principal component analysis (PCA). For the sake of brevity, we present here one 2D scatter plot for each oversampling technique, the one that achieved higher PS, as well as the SHAP [21] plot for WGAN.

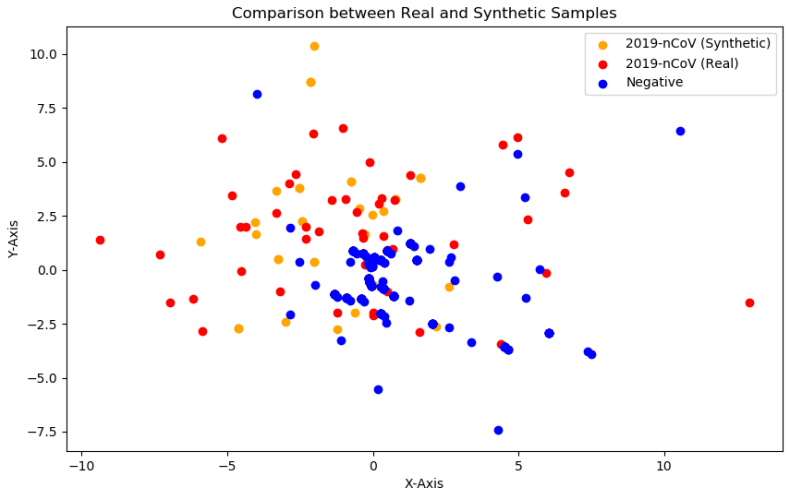

Fig. 1. Data from the scenario where 50% of the real samples were replaced by synthetic samples created with ROS

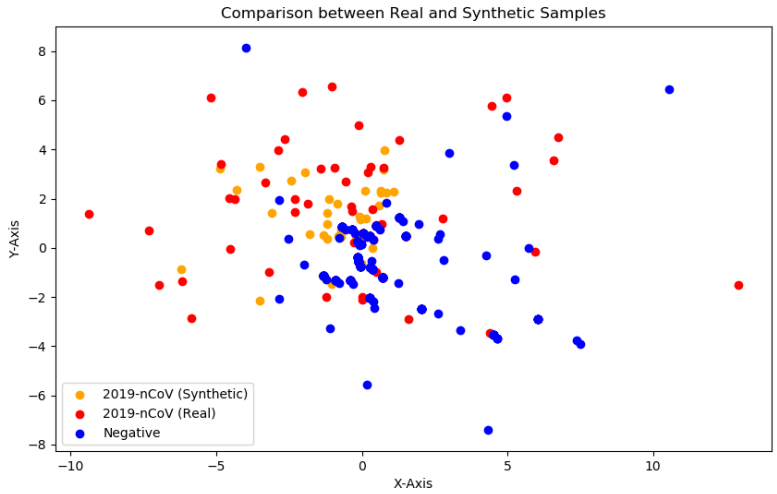

Fig. 2. Data from the scenario where 50% of the real samples were replaced by synthetic samples created with SMOTE

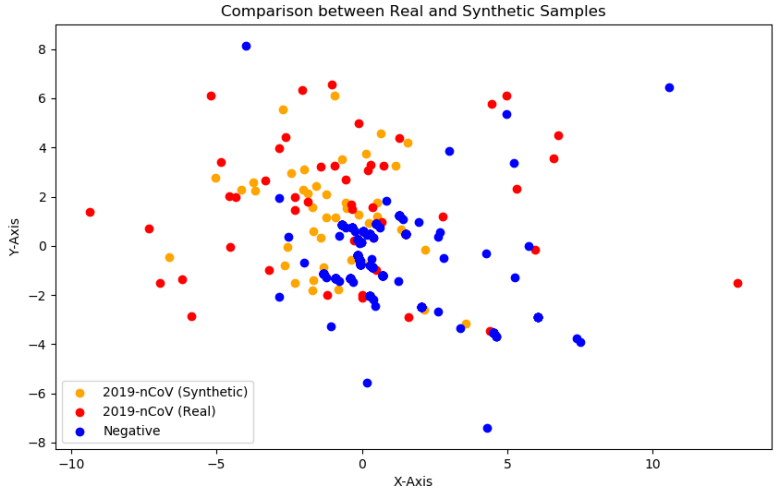

Fig. 3. Data from the scenario where 50% of the real samples were replaced by synthetic samples created with ADASYN

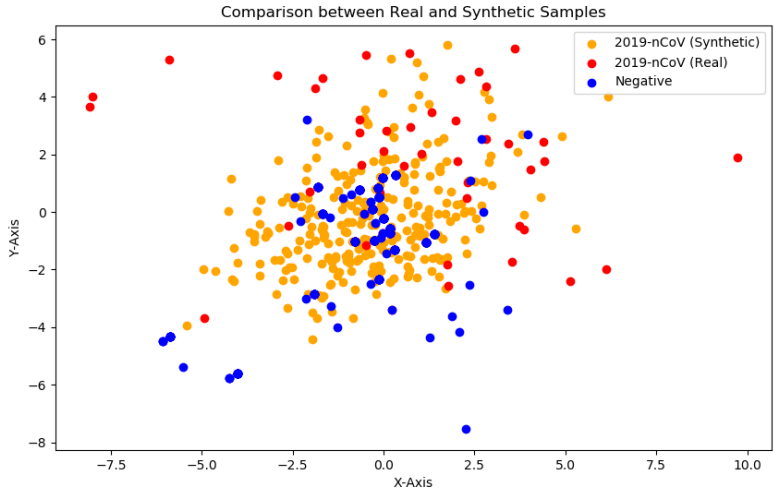

Fig. 4. Data from the scenario where 50% of the real samples were replaced by synthetic samples created with WGAN

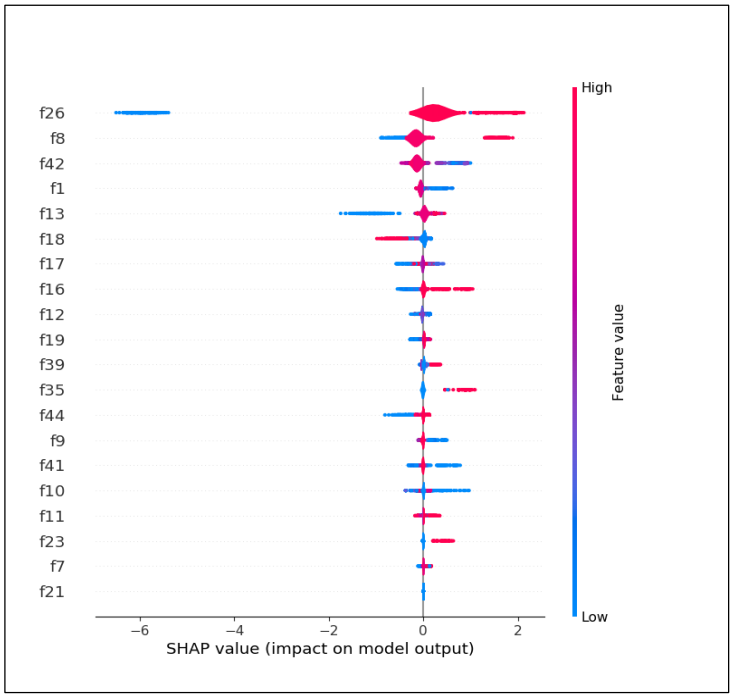

Fig. 5. The impact of each variable on model's output for the scenario in Fig. 4, as interpreted by Shapley Additive Explanations (SHAP) model [21]

After training the classifier with the scenarios in Table 1, optimizing the parameters in Table 2, we found the parameters that lead to the highest PS for each scenario, as described in Table 3, Table 4, Table 5, Table 6, and Table 7.

Table 3. Best hyperparameter values for the "No Synthetic Scenario"

Scenario	Hyperparameter	Value
NO SYNTHETIC (Training with 70%) (Test set: 782 samples)	n_estimators	83
	min_split_gain	0.15
	num_leaves	10
	learning_rate	0.246
	colsample_bytree	0.4
	boosting_type	"gbdt"
Precision	0.582	
Recall	0.759	
F1-Score	0.659	

Table 4. Best hyperparameter values for the (real +) ROS scenario

Scenario	Hyperparameter	Value
Real + ROS (trained with 50% synthetic positive samples)	n_estimators	10
	min_split_gain	0.20
	num_leaves	18
	learning_rate	0.008
	colsample_bytree	0.35

(Test set: 558 samples)	boosting_type	“gbdt”
Precision	0.584	
Recall	0.534	
F1-Score	0.558	

Table 5. Best hyperparameter values for the (real +) SMOTE scenario

Scenario	Hyperparameter	Value
Real + SMOTE (trained with 50% synthetic positive samples) (Test set: 558 samples)	n_estimators	47
	min_split_gain	0.15
	num_leaves	18
	learning_rate	0.019
	colsample_bytree	0.40
	boosting_type	“dart”
Precision	0.585	
Recall	0.591	
F1-Score	0.588	

Table 6. Best hyperparameter values for the (real +) ADASYN scenario

Scenario	Hyperparameter	Value
Real + ADASYN (trained with 50% synthetic positive samples) (Test set: 558 samples)	n_estimators	44
	min_split_gain	0.20
	num_leaves	16
	learning_rate	0.002
	colsample_bytree	0.40
	boosting_type	“gbdt”
Precision	0.589	
Recall	0.541	
F1-Score	0.564	

Table 7. Best hyperparameter values for the (real +) WGAN scenario

Scenario	Hyperparameter	Value
Real + WGAN (trained with 50% synthetic positive samples) (Test set: 558 samples)	n_estimators	66
	min_split_gain	0.20
	num_leaves	5
	learning_rate	0.208
	colsample_bytree	0.35
	boosting_type	“dart”
Precision	0.967	
Recall	0.107	
F1-Score	0.193	

In addition to these results, we plotted the value distribution for two of the most relevant features on these models, according to SHAP [21] qualitative measure. The distributions presented as boxplot charts are presented in Fig. 6 and Fig. 7. Worth mentioning that on these plots, "pos_samples" is the distribution for positive samples, "neg_samples" for negative samples, "wgan_samples" represent the value distribution for WGAN, "ada_samples" for ADASYN, "sm_samples" for SMOTE, and "ros_samples" for ROS.

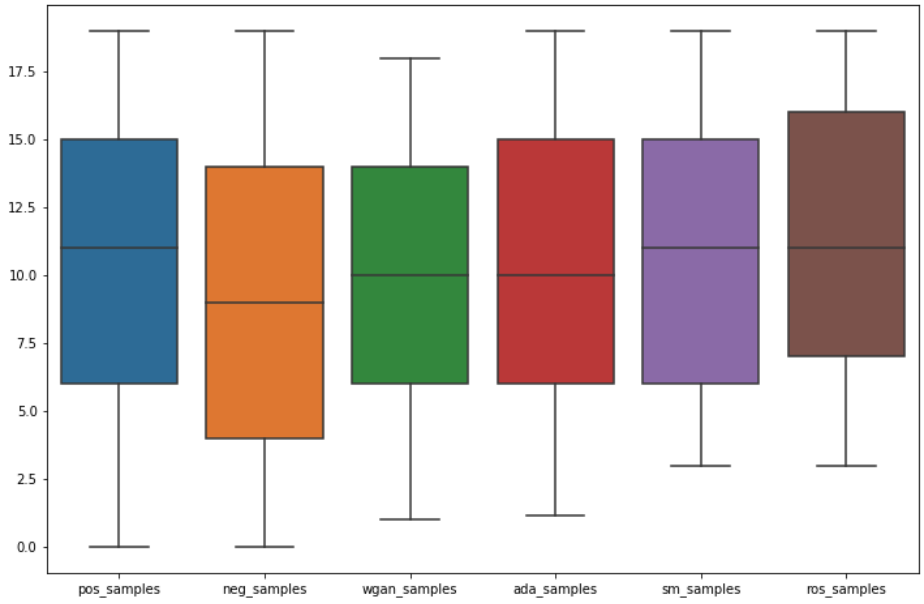

Fig. 6. Boxplot comparing the distribution of the different samples on feature "f1"

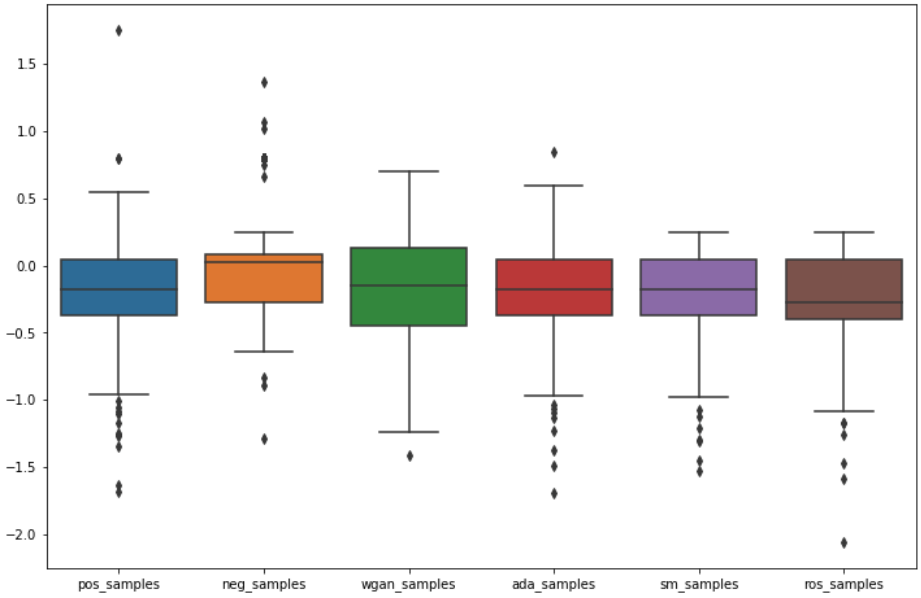

Fig. 7. Boxplot comparing the distribution of the different sample on feature "f8"

7. DISCUSSION

According to [26], machine learning-based models are of great interest in biomedical sciences given its capability to deal with high volume, high dimensional data, which are often generated in this domain. Especially in a pandemic scenario, it becomes even more important to deal with data with such characteristics, mak-ing machine learning an important tool to tackle the crisis. [27] points out that multidisciplinary research and international cooperation help speed up the development of potential solutions. A multidisciplinary approach can help with a fast-moving pandemic crisis using machine learning for patient outcome prediction [27]. It is the one this research aims to contribute.

As proposed by [27], we also believe that data for research on the pandemic crisis should be collected in scalable ways, making it easier for researchers to explore the data and generate insights. The data used for this research, for example, is one of the many datasets widely available online to promote the development of machine learning solutions to tackle the disease, and which required some preprocessing to be used as input for machine learning purposes [26].

Although the preprocessing steps are taken throughout this research severely decreased the number of available features used for augmentation and training of the final classifier, they are much relevant to prepare data for data mining and machine learning problems as suggested by [25]. A combination of sparse-learning-based models such as RFE and statistical-based techniques for feature elimination are also evaluated by [25] in different datasets. Both are found to be beneficial to decrease model complexity.

As far as synthetic samples are concerned, a recent study [16] compared SMOTE and ADASYN, other oversampling techniques in different toy datasets, and their final performance measure by a classifier f-measure are fairly similar. This is also something found in Table 4 to Table 6, mainly in terms of PS. In fact, by analyzing Fig. 2 and Fig. 3, it is also possible to see such a pattern. ROS (Table 4, Fig. 1), although being the simplest oversample technique [9], presented results much similar to SMOTE and ADASYN in similar training scenarios, which points to the fact the SMOTE and ADASYN might have created good quality synthetic samples, but without much novelty in this particular dataset.

The samples generated with WGAN, as seen in Fig. 6 and Fig. 7, comply with the findings in [5][6], as well as [18] and [19], which points to the high-quality of samples created with such technique. Fig. 4 demonstrates that such synthetic samples are much different from the ones in Fig. 1 to Fig. 3 and, therefore, are presenting enough novelty to the classifier that it is even able to more precisely identify TP on the test set, as demonstrated on PS in Table 7. Fig. 5 also demonstrates that not a single feature is responsible for the result achieved with WGAN, but a combination of relevant features explored in Fig. 6 and Fig. 7.

Due to the high amount of model optimization techniques used in this research and the computing power needed to generate viable WGAN samples, we believe it might not be possible to work on retraining the WGAN model in a real-time fashion. Additionally, since this model is more specific than sensitive, it would be more reliable as a secondary diagnostic measure, meaning it would be used for those patients initially selected by a sensitive measure on an initial screening. Therefore, a system containing this technique could be retrained from time to time, with new patients being tested with the most updated model by the time their exam results are ready.

8. CONCLUSION

From these experiments, it was possible to identify that synthetic samples created with WGAN offer an alternative to traditional data augmentation techniques for oversampling a minority class. Additionally, the experiments showed that a classifier trained with samples created with WGAN outperform the precision of all other scenarios, including those where the classifier was

trained with samples created from other oversampling strategies. Given this dataset's nature, we understand that greater precision is preferable as an additional detection mechanism for 2019-nCoV.

By analyzing the scatter plots, we were also able to notice that positive and negative samples frequently overlap, which means that the features available are not capable of clearly discriminating between positive and negative samples in a 2-D space. It is also possible to notice that in Fig. 4, samples created with WGAN occupy a wide range of possible positions on the plot, rarely overlapping, some-thing that is very different from the results observed in other oversampling algorithms. We understand that this represents the realistic nature of these samples, as also demonstrated in [5][6][7]. Finally, we believe that oversampling based on generative adversarial networks should be further explored in other datasets to tackle issues with different configurations, such as time-series.

REFERENCES

1. do Brasil, Ministério da Saúde., Acurácia dos diagnósticos registrados para COVID-19, 2020, pp.2-20
2. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative Adversarial Nets. In: Advances in Neural Information Processing Systems; 2014. pp. 2672–2680
3. Han, Z., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.: StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence. 2019, vol. 41, issue 8, pp. 1947-1962.
4. Wang, H., Qin, Z., Wan, T.: Text Generation Based on Generative Adversarial Nets with Latent Variable, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2018, vol 10, pp. 92-103.
5. M. Arjovsky, S. Chintala, Bottou, L., Wasserstein Generative Adversarial Networks, 34th International Conference on Machine Learning, ICML 2017, vol 1, pp.298-321
6. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A., Improved Training of Wasserstein GANs, Advances in Neural Information Processing Systems, 2017, pp. 5768-5778
7. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W., SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research, 2006, issue Sept. 28, pp. 321-357
8. Haibo, H., Bai, Y., Garcia, A., Li, S., ADASYN: Adaptive synthetic sampling approach for imbalanced learning, Proceedings of the International Joint Conference on Neural Networks, 2008,pp. 1322-1328
9. Ye Li, Y C., Zheng, Z., Oversampling methods for imbalanced classification, Computing and Informatics, 2015, v.34, issue 5, pp. 1017-1037
10. Riou, J., Althaus, C., Pattern of early human-to-human transmission of Wuhan 2019 novel coronavirus (2019-nCoV), December 2019 to January 2020, 2019, Eurosurveillance, v. 25, issue 4, pp.1-5
11. Roh, Y., Heo, G., Whang, S.E., A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective, IEEE Transactions on Knowledge and Data Engineering, 2019, pp.1-20
12. Bullock, J., Luccioni, A., Pham, K. H., Nga Lam, C. S., Luengo-Oroz., M., Mapping the Landscape of Artificial Intelligence Applications against COVID-19, 2020, pp.1-32
13. Wong, S., Gatt, A., Stamatescu, V., Understanding Data Augmentation for Classification: When to Warp?, 2016 International Conference on Digital Image Computing: Techniques, pp.1-6

14. Wang, J., Perez, L., The Effectiveness of Data Augmentation in Image Classification using Deep Learning, Stanford University research report, 2017, pp.1-8
15. Sefara, T., Marivate, V., Improving short text classification through global augmentation methods, 2019, pp.1-15
16. Amin, A., Anwar, S., Adnan, A., Nawaz, M., Horward, N., Qadir, J., Hawalah, A., Hussain, A., Comparing Oversampling Techniques to Handle the Class Imbalance Problem: A Customer Churn Prediction Case Study, 2016, IEEE Access, vol. 4, issue Ml, pp. 7940-7957
17. Creswel, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A., Generative Adversarial Networks: An Overview, IEEE Signal Processing Magazine, 2017, pp.1-14
18. Nguyen, T., Nguyen C., Nguyen, D., Nguyen D. T., Deep Learning for Deep Fakes Creation and Detection, 2019, pp.1-14
19. Fabbri, C., Conditional Wassertein Generative Adversarial Networks, University of Minnesota, 2018, pp.1-11
20. Flach, P., Kull, M., Precision-Recall-Gain Curves: PR Analysis done right, CONFERENCE and Workshop on Neural Information Processing Systems, NIPS, 2015, pp.1-9
21. Lundberg, S., Lee, S., A Unified approach to Interpreting Model Predictions, CONFERENCE and Workshop on Neural Information Processing Systems, NIPS, 2017, pp.1-10
22. Kaggle, Diagnosis of COVID-19 and its clinical spectrum, https://www.kaggle.com/einsteindata4u/covid19, last accessed 2020/04/25
23. LightGBM, Parameters Tuning, https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html, last accessed 2020/05/13
24. GANify Python Library, Github, https://github.com/arnonbruno/ganify, last accessed 2020/05/13
25. Jundong, L., Kewei, C., Suhang, W., Morstatter, F., Trevino, R., Tang, J., Liu H., Feature Selection: A Data Perspective, 2018, ACM Comput. Survev. 50, 6, Article 94, pp.12-70, DOI:https://doi.org/10.1145/3136625
26. Sajda, P., Machine Learning for Detection and Diagnosis of Disease, Annual Review of Biomedical Engineering, 2006, pp. 2-30.
27. Bullock, J., Luccioni, A., Pham. K, Sin Nga Lam, C., Luengo-Oroz, M., Mapping the Landscape of Artificial Intelligence Applications Against COVID-19, 2020, pp.2-32
28. Guollin, K., Meng, K., Finley, T., Wang, T., Chen, W., Weidong, M., Ye, Q., Tien-Tan, L., LightGBM: A Highly Efficient Gradient Boosting Decision Tree, CONFERENCE and Workshop on Neural Information n. 4, p. 42-47, 2012.