GENERATIVE ADIVERSARIAL NETWORKS TO VALIDATE DATA IN DISCRETE EVENT SIMULATION

Gustavo Teodoro Gabriel
gustavo.teodoro.gabriel@gmail.com
Afonso Teberga Campos
afonso.teberga@gmail.com
José Arnaldo Barra Montevechi
montevechi@unifei.edu.br
Fabiano Leal
fleal@unifei.edu.br

Universidade Federal de Itajubá –UNIFEI
Avenida BPS, 1303, Itajubá –MG

RESUMO

A validação computacional é fundamental para que os resultados da simulação sejam efetivos no sistema real. A literatura sugere o uso de técnicas estatísticas para essa comparação, porém, muitas vezes os pressupostos estatísticos são violados. Assim, as Redes Adversárias Geradoras (GANs) são uma alternativa, pois elas se adaptam a qualquer tipo de dados. O trabalho tem como objetivo utilizar as GANs para gerar dados sintéticos a partir dos dados reais e usar o discriminador para discriminar dados reais de simulados. Foi realizado o treinamento de cinco distribuições estatísticas e submetido 10000 distribuições com as mesmas características para testar o poder do teste. Além disso, um caso real de simulação a eventos discretos em um centro de atendimento de pacientes com suspeita de COVID-19 foi aplicado a nova técnica de validação. Os resultados mostram que as GANs são efetivas ao discriminar dados e podem ajudar na validação de modelos computacionais.

PALAVRAS CHAVE. SED. GANs. Validação.
Área Principal: IC – Inteligência Computacional/ SIM – Simulação

ABSTRACT

Computer validation is essential for the results of the simulation to be effective in the real system. The literature suggests the use of statistical techniques for this comparison, however, statistical assumptions may be violated. Thus, Generative Adverse Neural Networks (GANs) are an alternative since they adapt to any data. The work aims to use GANs to generate synthetic data from the real data and use the discriminator to discriminate real from simulated outputs. Five statistical distributions were trained and 10,000 distributions with the same characteristics were submitted to verify the power test. In addition, a real case of a Discrete Event Simulation in a hospital that assists patients with suspected COVID-19 was applied to the new validation technique. The results show that GANs are effective in discriminating data and can help in the validation of computer models.

KEYWORDS. DES. GANs. Validation.
Main area: IC – Computer Intelligence/ SIM – Simulation

1. Introduction

Modeling and Simulation include methods, tools, and techniques to represent, experiment, and study complex systems [Scheidegger et al. 2018] seeking to improve and optimize processes. According to [Goldsman 2007], simulation is the most useful tool in industrial engineering, operational research, and management science. Thereby, simulation models are often used to solve problems and assist in decision-making, especially in situations that involve personal and/or financial risk [Sokolowski and Banks, 2010]. For this reason, modelers and decision-makers are concerned with reliability and results. In this sense, decisions must be made based on correct simulation results, therefore, the model must represent the real world [Popovics et al. 2016], in other words, the model must be valid.

[Zeigler and Nutaro, 2016] consider validity as the degree that a model represents its system counterpart. The validation processes aim to quantify the accuracy of the model, comparing the simulation results with experimental or operational results in the real world [Thacker et al. 2004], determining whether the model adequately represents the system for the desired purpose [Law 2015], [Sargent and Balci 2017]. Although it is not possible to guarantee that a model is 100% valid, [Olsen and Raunak, 2015] say that validation is a crucial part of the results obtained from the simulation model to be reliable.

All output metrics related to the model's objectives must be validated. The literature presents several validation techniques, despite this, [Sargent 2013] states that statistical comparisons between the simulated model and the real system are preferable for presenting an objective approach and mathematical evidence. However, in some models, the data may be automatically correlated or assumptions of normality and sample size are violated, preventing the use of classical statistical techniques [Comas et al. 2008]. In order to overcome this issue, Generative Adversarial Networks (GANs) may be used to train data and discriminate them. They were developed by [Goodfellow et al. 2014] and are characterized as models of Deep Learning (DL). In this sense, the article aims to propose a validation method for Discrete Event Simulation models using GANs in data validation. Then, we propose a new approach and tested it in theoretical distributions.

The article is divided as follows: section 1 presented the introduction and the problem context. Section 2 presents a theoretical background about Discrete Event Simulation and GANs. Then, in section 3, there is the method proposed for the validation. Section 4 presents the cases, analysis, and results. Finally, in section 5, the conclusions of the work are shown.

2. Theoretical background

2.1. Discrete Event Simulation and model validation

According to [Bateman et al. 2013], Discrete Event Simulation (DES) is defined as the simulation in which the variables change instantly in a space of time, occurring in a single moment and oriented by events. Thus, the model presents its changes at random, where the times of the events are the points of the simulated time [Schriber et al. 2017]. DES incorporates effects of system variability when using probability distributions to characterize the variables of the uncertainty system [Botín et al. 2015].

[Skoogh et al. 2012] claim that DES is a promising tool for planning, designing and improving process flows. [Jahangirian et al. 2010] say that it is appropriate for tactical and operational levels in decision-making. According to the authors, it tends to be convenient for detailed analysis of processes, use of resources, queuing and short-term analysis.

Although the tool helps managers to improve and plan the systems, it only presents real impacts if the simulated model reflects the real process. The step is called model validation. In this sense, the literature offers a number of techniques to carry out the model validation. However, [Sargent 2013] states that statistical comparisons between the simulated model and the real system are preferable for presenting an objective approach and mathematical evidence.

Nevertheless, in the case of models with high variability, the data can be automatically correlated or assumptions of normality and sample size can be violated, preventing the use of classical statistical techniques [Comas et al., 2008]. Artificial Intelligence may overcome the problem.

2.2. Generative Adversarial Networks

"Deep fake" are synthetic contents generated through DL. The term was created in 2017, after a Reddit platform user, called "deep fakes", claimed to have developed a Machine Learning (ML) algorithm that could replace faces in adult content videos with the faces of celebrities. According to [Tolosana et al. 2020], in the following years, synthetic contents were used for other harmful purposes, such as fake news and financial fraud, causing considerable concern on authorities around the world.

Although the data were used to cause harmful situations, they are also used for positive content, especially in the healthcare area. For example, [Sorin et al. 2020] review studies that aim to generate realistic medical imaging results. According to the authors, the images do not violate patient confidentiality because the images generated by the ML algorithm are not real. Moreover, the images may be used for academic purposes and for training other ML models. The authors also highlight the use of "deep fake" to complete or refine existing medical images.

In this sense, the Generative Adversarial Networks (GANs) can generate synthetic data close to real. They are an AI technique proposed by [Goodfellow et al. 2014] and aims to generate realistic (but usually not the same) data. GANs comprise two adversarial Artificial Neural Networks (ANN), which are trained iteratively and compete against each other: the data Generator *(G)* and the data Discriminator *(D)* [Pan et al., 2019]. *G* is a differentiable ANN (of parameters θg) that learns to generate synthetic samples p_g by mapping a latent input variable *z* (a noise, with no practical meaning) to the real data space $G(z,\theta_g)$. $D(x,\theta_d)$ is also a differentiable ANN (of parameter θ_d) which outputs a single scaler representing the probability that an observation *x* will originate from the real data and not from p_g [Goodfellow et al. 2014].

D is trained with access to real data and synthetic observations generated by *G*. The training aims to maximize the probability of classifying both synthetic and real observations correctly. On the other hand, training *G* aims to minimize the probability of *D* classifying synthetic data as false, minimizing $log(1–D(G(z, \theta_g),\theta_d)$. After some iterations, *G* starts to produce synthetic samples similar to the real ones and the discriminator *D* finds it more difficult to differentiate them (Brownlee, 2020). If the discriminator D does not differentiate the data, it concludes that the generator was able to achieve its goal and learned the real data distribution [Pan et al. 2019]. Figure 1 shows how the GANs' training is performed for image generation.

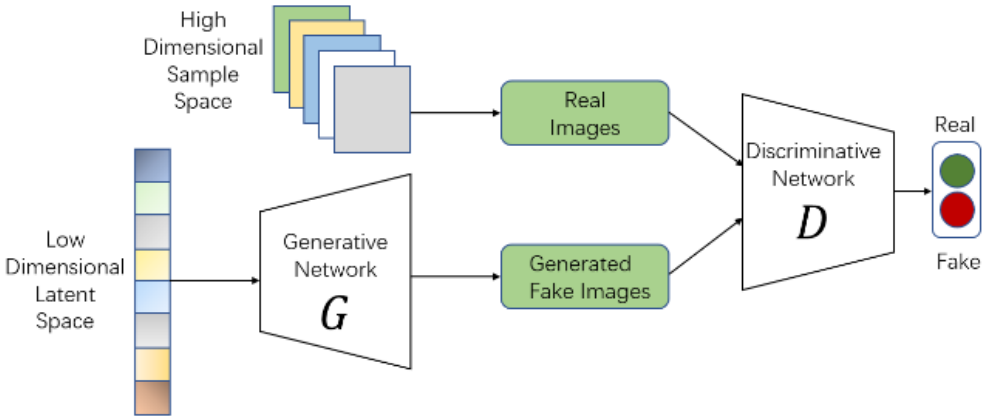


Figure 1: GANs' architecture. Adapted from [Cai et al. 2020]

The GANs have been used in several areas with great success, especially to synthesize medical images and increase sample sizes [Yi et al. 2019] and in the computer vision area [Sorin et al. 2020]. Although the GANs presented more influence in images, videos, and sound processing, they are able to learn complex distributions that represent the behavior of a population and can generate new samples following the same fit [Goodfellow et al. 2014].

Moreover, after the training, the discriminator can discriminate whatever kind of data. [Goodfellow et al. 2014] and [Pan et al. 2019] say that, when the training is perfect, the discriminator does not know how to identify which values are real and which values are synthetic. In this sense, the discriminator evaluates as a random probability, with a 50% chance of the data being true or false. However, as there is a double set of networks, it is not always easy to generate data that can reach this level of precision [Bronwlee 2020]. Then, for the purpose of the present work, the discriminator needs to discriminate the data between 47.5% and 52.5%. In this approach, the modeler can validate simulated and real data.

3. Framework to validation using GANs

The aim of data validation through the GANs is to compare the judgment by the discriminator between the simulated and the real data. The framework is divided into two steps: the training and the testing step, as shown in Figure 2. All steps were coded in Python using TensorFlow, Keras, and scikit-learn libraries.

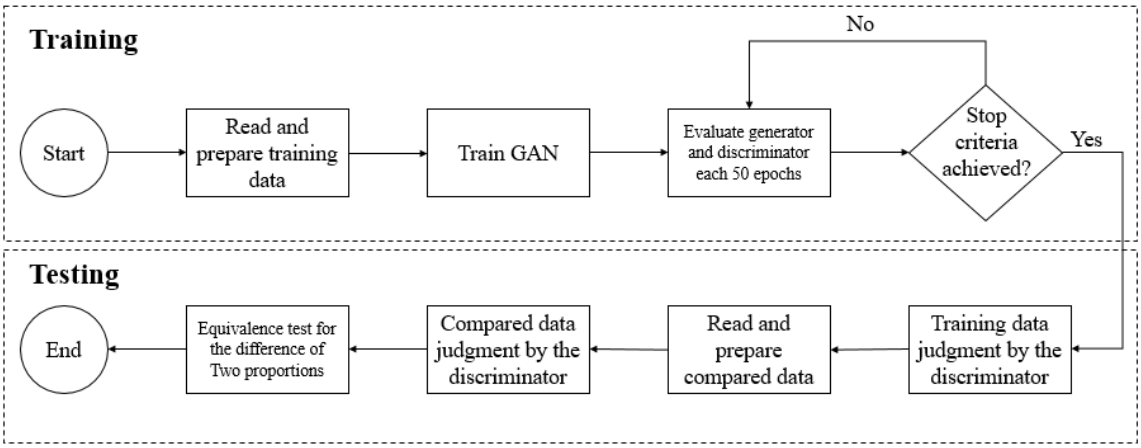


Figure 2 – Framework to validation using GANs

In the first step, the algorithm imports input data from the source database. Each observation from a specific item has to be in the same row while columns contain sample values *(x)*. If the modeler wants to model a joint distribution, all observation attributes in a row must be related to the same observation. In this study, we assume that the data is sufficient and it is not necessary to replace missing values, clean or evaluate outliers. The data are scaled using robust scaling [Raju et al. 2020] because it leads to more effective ML [Feng 2021].

In the second step, the algorithm trains the GANs. The discriminator access batches of synthetic data and real data of equal size and try to correctly classify the observations. Then, the generator is also trained to generate batches of synthetic data that confuse the discriminator. This interaction of the adversarial learning process is repeated until there are no remaining batches of real data, completing a learning epoch. The maximum number of epoch is 1000.

After completed the epoch training and every 50 epochs, the algorithm is evaluated. The evaluation is made to guarantee that the synthetic samples correspond to the training data. In this sense, the study uses the k-NN-based Classifier Two-Sample Tests (C2ST) [David and Oquab, 2017] and [Cai et al. 2019] to classify the data. In k-NN, the observation class is determined by votes of the *k* nearest neighbors [Cover and Hart 1967]. Then, the classifier receives the dataset

with synthetic data and (all) training data, both scaled and in the same proportions, and tries to separate them. According to [David and Oquab], if the synthetic data are perfectly realistic, the k-NN algorithm classify each observation at random and the accuracy (A_C) is 50%. On the other hand, if synthetic data are not realistic at all, the classifier can easily separate the observations, and the expected A_C reaches 100%. However, data can be in a tolerance space and a one-proportion test is performed. The k-NN result should be inside the tolerance rate.

Moreover, the GANs stop to train in another condition. According to [Brownlee 2020] the discriminator evaluates randomly the training data if the generator can trick the discriminator. The condition happens, in other words, the generator tricks the discriminator if the discriminator discriminates the data around 50%. However, we are aware that the situation sometimes is not possible. In this sense, the GANs are trained until the discriminator discriminates data between 45.0% and 55.0%.

When the stop criteria are achieved, the discriminator judge the training data. Then, the compared data are inserted into the program to be prepared. In this step, the data need to be in the same way that the judgment data was inserted. As soon as the data are inserted, they are rescaled based on the trained data and then, the discriminator them.

In the final step, an Equivalence test for the difference of two proportions is carried out. The step comprises in verify statistically if the trained and compared data are equals. In addition, in this step, the modeler can choose how different the compared data may differ from the trained data, that is, there may be a previously determined tolerance.

The Equivalence test for the difference of two proportions measures whether the difference between the classifications of two populations is between an equivalence interval, in other words, between a tolerance limit. The approach used is the “Two one-sided test” (TOST), two unilateral tests created by [Schuirmann 1987]. Equation 1 shows the null (H_0) and alternative (H_1) hypothesis [Silva et al. 2009]:

$$H_{01}: p_1 - p_2 < -\delta \text{ or } H_{02}: p_1 - p_2 > \delta$$
$$H_1: -\delta \leq p_1 - p_2 \leq \delta \quad (1)$$

As mentioned before, [Lopez-Paz and Oquab, 2017] say that the C2ST aims to test whether two samples P and Q are classified equally or differently. Thus, when classifying a data set, the authors state that using the central limit theorem, the classification distribution follows a normal distribution, as shown in Equation 2:

$$f(x) \sim \mathcal{N}\left(\bar{p}, \frac{\bar{p}(1-\bar{p})}{n_{te}}\right) \quad (2)$$

Where:
f (x) is the classification distribution of the data;
$\bar{p}$ is the data average classification;
n_{te} is the number of data in the sample.

4. Analysis and Results

4.1. Training

We trained five statistical distributions following the training steps presented in section 3, as presented in Table 1. The training phase was carried out with three different sample sizes: 10000, 5000, and 2000. The training was performed for three amount of data in the sample size to measure how many epochs is necessary to achieve the conditions and the effect in the power test.

Table 1 – Distributions used for train and power test

Distribution	Parameters	Correlation
Normal (a)	Mean = 100; Std. dev = 3	-
Bivariate Normal (b)	Mean = [100, 100] Cov = [[9, 2.4], [2.4, 9]]	0.8
Bivariate Normal (c)	Mean = [100, 100] Cov = [[9, -2.4], [-2.4, 9]]	-0.8
Bivariate Normal (d)	Mean = [100, 100] Cov = [[9, 0], [0, 9]]	0.0
Multivariate Normal (e)	Mean = [100, 100, 100] Cov = [[9.00, 7.65, 4.50], [7.65, 9.00, 3.60], [4.50, 3.60, 4.00]]	$x_1x_2 = 0.85$ $x_1x_3 = 0.75$ $x_2x_3 = 0.60$

The minimal level of accuracy desired is 95.0%. In this sense, after the training, the lower accuracy reached was by a bivariate normal (positive correlation) with 10000 data (95.8%) while the higher accuracy was gotten by a multivariate normal with 5000 data. Also, there is a pattern in the training. In general, if the training phase presents more data, it used to reach the conditions in less epoch than when training with fewer input data quantity. Table 2 shows the accuracy and the number of epochs necessary to achieve the stop condition while Figure 3 present the training process for a normal distribution with 10000 data (a) and a multivariate normal with 5000 data (b).

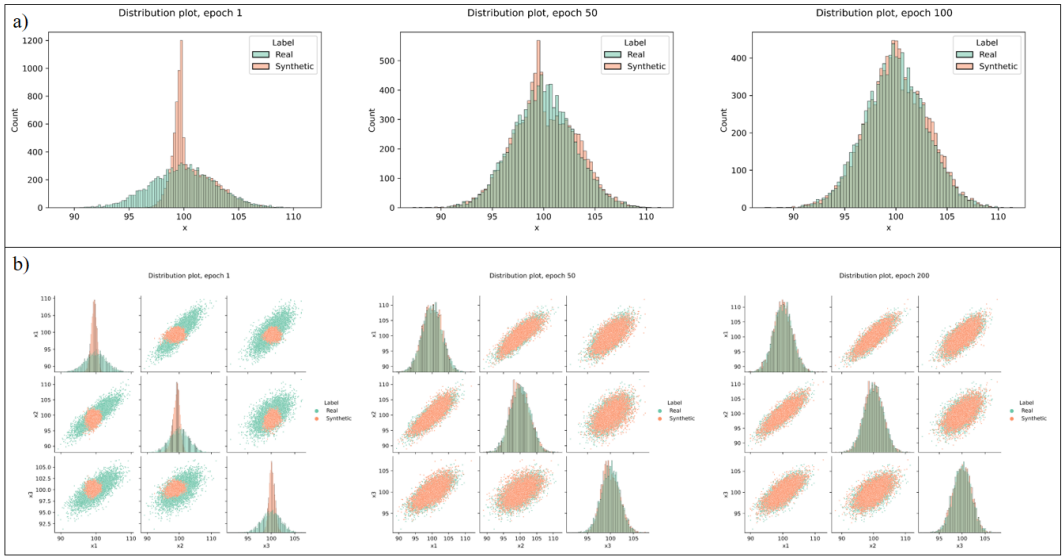


Figure 3 – Training process for normal (a) and multivariate normal (b)

Table 2 – Distributions used for training and power test

Input data	**10000**		**5000**		**2000**	
Distribution	**Epoch**	**Accuracy**	**Epoch**	**Accuracy**	**Epoch**	**Accuracy**
(a)	100	97.50%	200	96.60%	550	98.00%
(b)	150	95.80%	550	97.40%	500	98.00%
(c)	450	96.70%	450	97.80%	250	97.50%
(d)	250	97.30%	350	97.50%	400	98.00%
(e)	250	97.40%	200	99.10%	300	96.70%

4.2. Power Test

Since the statistical tests are subject to errors and, to show how assertive is the proposed approach, it was carried out some tests with the discriminator for each trained distribution. As an example, we choose the normal distribution. As mentioned before, it was trained with a sample of 10000 in the training data and obtained 97.5% of accuracy in 100 epochs.

After the training phase, it was generated 10000 distributions with the same behavior as the input data, that is, a normal with a mean of 100 and a standard deviation of 3. In addition, we choose a tolerance in the judgment of 5.0%. In the first moment, the generated distributions presented 10000 data. All of them were submitted to the discriminator that had to indicate that the training data and compared data were statistically significant. The process was repeated with the same distribution, however, with 5000, 2000, 1000, 100, and 10 data. In this sense, we can obtain the power test for a normal distribution. Again, the process was repeated with a tolerance of 10.0%. Table 2 shows the results.

Table 2 – Power test – Normal distribution

N. of input data	**10000**		**5000**		**2000**	
N. of data in distribution	**Tolerance**		**Tolerance**		**Tolerance**	
	5%	**10%**	**5%**	**10%**	**5%**	**10%**
10000	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
5000	100.00%	100.00%	100.00%	100.00%	99.03%	100.00%
2000	99.08%	100.00%	98.78%	100.00%	86.58%	100.00%
1000	83.76%	100.00%	84.03%	100.00%	63.72%	99.98%
100	0.00%	32.03%	0.00%	23.11%	0.00%	23.27%
10	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

We observed that with 5000 data or more in the compared data, the discriminator presented an assertiveness of 100.00%, regardless of the chosen tolerance. Nevertheless, when the number of data in the discriminator gets to be judged decreases, it features a lower power test. It can also be seen that, by providing less training data, the power test also begins to decrease when judging 2000 data or less. On the other hand, if the tolerance is 10.0%, the discriminator can differentiate the distribution in all most cases for at least 1000 compared data. This pattern can also be observed for the judgment of bivariate data for the positive correlation (Table 3), negative correlation (Table 4), and without correlation (Table 5).

Table 3 – Power test – Bivariate normal distribution (correlation 0.8)

N. of input data	**10000**		**5000**		**2000**	
N. of data in distribution	**Tolerance**		**Tolerance**		**Tolerance**	
	5%	**10%**	**5%**	**10%**	**5%**	**10%**
10000	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
5000	100.00%	100.00%	100.00%	100.00%	99.77%	100.00%
2000	98.25%	100.00%	98.66%	100.00%	92.20%	100.00%
1000	82.66%	100.00%	81.40%	99.99%	68.29%	99.99%
100	0.00%	23.38%	0.00%	31.01%	0.00%	24.03%
10	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Table 4 – Power test – Bivariate normal distribution (correlation -0.8)

N. of input data	10000		5000		2000	
N. of data in distribution	Tolerance		Tolerance		Tolerance	
	5%	10%	5%	10%	5%	10%
10000	100.00%	100.00%	100.00%	100.00%	99.69%	100.00%
5000	100.00%	100.00%	99.77%	100.00%	94.10%	100.00%
2000	98.00%	100.00%	90.91%	100.00%	74.13%	100.00%
1000	81.89%	100.00%	66.89%	99.99%	50.72%	99.97%
100	0.00%	31.75%	0.00%	22.36%	0.00%	22.02%
10	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Table 5 – Power test – Bivariate normal distribution (correlation 0.0)

N. of input data	10000		5000		2000	
N. of data in distribution	Tolerance		Tolerance		Tolerance	
	5%	10%	5%	10%	5%	10%
10000	100.00%	100.00%	100.00%	100.00%	99.74%	100.00%
5000	100.00%	100.00%	99.75%	100.00%	96.44%	100.00%
2000	96.25%	100.00%	91.13%	100.00%	77.60%	100.00%
1000	75.64%	100.00%	68.56%	99.98%	55.19%	99.97%
100	0.00%	22.98%	0.00%	22.91%	0.00%	22.91%
10	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Table 6 shows the result for multivariate normal distribution. Although the power tests present the same behavior as a bivariate distribution, the results indicate that, when the number of variables related to a specific element starts to increase, more input data are necessary to the training phase. Indeed, when the training is done with 2000 data and judged with 5000 for a 5% tolerance, the model showed an accuracy of 86.58% for the univariate distribution, 81.31% for the bivariate distribution, and dropped dramatically to 53.57% for the multivariate distribution.

Table 6 – Power test – Multivariate normal distribution (positive correlation)

N. of input data	10000		5000		2000	
N. of data in distribution	Tolerance		Tolerance		Tolerance	
	5%	10%	5%	10%	5%	10%
10000	100.00%	100.00%	100.00%	100.00%	96.67%	100.00%
5000	99.71%	100.00%	99.51%	100.00%	88.58%	100.00%
2000	88.98%	100.00%	73.23%	100.00%	53.57%	100.00%
1000	64.04%	99.99%	48.18%	99.82%	32.45%	99.72%
100	0.00%	21.38%	0.00%	28.31%	0.00%	24.12%
10	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

4.3. Model validation

In order to test the validation in a real model, we applied the technique in a planned hospital that is used to assist patients with suspect of COVID-19. The patients arrive in the hospital; they are registered and go to triage. In the triage, they receive six classifications: white,

blue, green, yellow, orange, and red. After the registration, they are attended by a doctor and can be discharged, take some medicine or perform an x-ray or go to hospitalization. We considered for the validation the time between the patient arrives in the hospital until be attended by the doctor (DTD) and the length of stay (LOS) from patients that do not go to the hospitalization. According to Table 3, 4, and 5, when we are evaluating two characteristics, it is necessary at least 5000 data in the training test and at least 5000 data for the judgment to obtain a 100% of success. Then, the training phase received 6713 data, while the testing received 6701. Figure 4 shows the training process for the data.

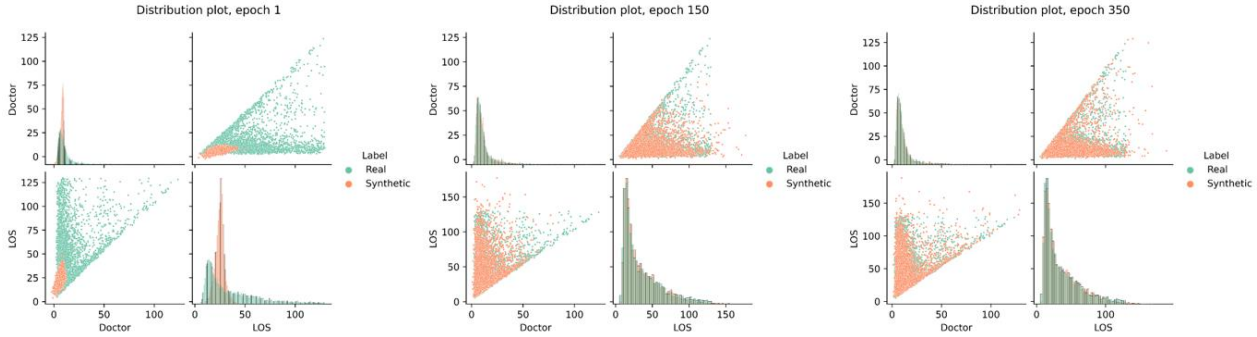


Figure 4 – Training process for time from DTD and LOS

The model obtained an accuracy of 96.2% in 350 epochs. Moreover, the discriminator judge the data in 53.81%. After the training phase, data were compared and there was evidence (p-vale < 0.05) to claim equivalence, as shown in Figure 5.

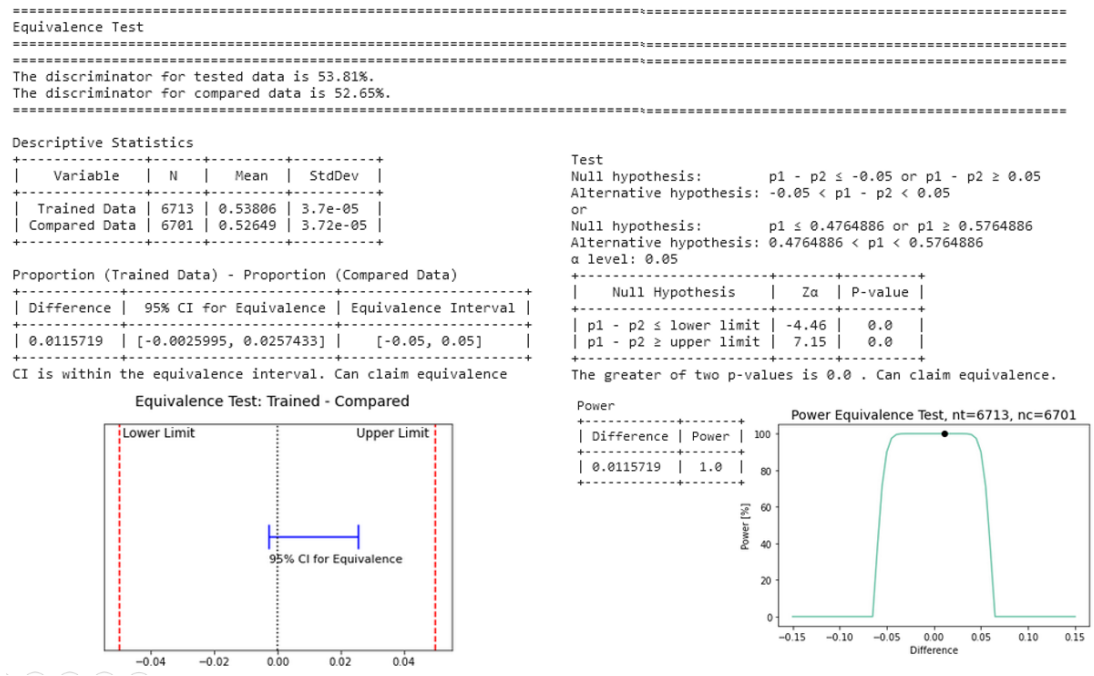


Figure 5 – Equivalence Test for Difference of Two Proportions using GANs

5. Conclusions

This study aimed to analyze how GANs can help discriminate data since they can adapt to any kind. Then, we proposed a new approach divided into two phases, where the first one

comprises in to generate the data until the artificial networks achieve the necessary accuracy defined by the modeler and the discriminator judge the data between 45.0% and 55.0%. The second phase comprises in to submit the training data to be judge and after it is performed a Test for Difference of Two Proportions.

Since the statistical tests are subject to errors and, to show how assertive is the proposed approach, it was carried out 10000 tests with the discriminator for a normal distribution, a bivariate normal distribution with positive, negative, and without correlation, and a normal multivariate. All of them were trained with a sample of 10000 in the training data and after that, it was generated 10000 distributions with the same behavior as the input data. The tolerance in the judgment was 5.0%. In the first moment, the generated distributions presented 10000 data. All of them were submitted to the discriminator that had to indicate that the training data and compared data were statistically significant. The process was repeated with the same distribution, however, with 5000, 2000, 1000, 100, and 10 data. Again, the process was repeated with a tolerance of 10.0%. Finally, we tested a real model, testing the validation of a hospital that attends patients' suspect of COVID-19.

The results showed that GANs are efficient to discriminate data and can help in the model validation phase in Discrete Event Simulation. However, according to the study, it is necessary more than 5000 data in both phases to achieve confinable results. For future works, we suggest training the chosen distribution with different seeds to obtain a confidence interval. Moreover, we suggest evaluating other distributions to see how GANs can discriminate them.

Acknowledgments

The authors thanks Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), and Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) for the support.

References

Bateman, R.E.; Bowden, R.O.; Gogg, T.J.; Harrel, C.R.; Mott, J.R.A.; Montevechi, J.A.B. (2013). *Simulação de sistemas: aprimorando processos de logística, serviços e manufatura.* 1. ed. Rio de Janeiro: Elsevier.

Botín, J.A.; Campbell, A.N.; Guzmán, R. (2015). A discrete-event simulation tool for real-time management of pre-production development fleets in a block-caving project. *International Journal of Mining, Reclamation and Environment*, 29(5): 347-356.

Brownlee, J. (2020). *Adversarial Networks with Python: Deep Learning Generative Models for Image Synthesis and Image Translation*. Machine Learning Mastery.

Cai, H.; Goggin; B., Jiang; Q. (2019). Two-sample test based on classification probability. *Statistical Analysis and Data Mining*, 13(1):5-13.

Comas, M.; Castells, X.; Hoffmeister, L.; Román, R.; Cots, F.; Mar, J.; Gutiérrez-Moreno, S.; Espallargues, M. Discrete-event simulation applied to analysis of waiting lists. (2008). Evaluation of a prioritization system for cataract surgery. *Value Health*, 11(7):1203-1213.

Cover, T.; Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21-27.

David, L.-P.; Oquab, M. (2017). Revisiting Classifier Two-sample Tests. In: *Proceedings of the 2017 International Conference on Learning Representations*, 1-15.

Feng, R. (2021). Improving uncertainty analysis in well log classification by machine learning with a scaling algorithm. *Journal of Petroleum Science and Engineering*, 196.

Goldsman, D. (2007) Introduction to simulation. In: *Winter Simulation Conference, Proceedings*, pp. 26-37.

Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. (2014). Generative Adversarial Nets. *Advances in Neural Information Processing Systems (NIPS)*, pp. 1-9.

Jahangirian, M.; Eldabi, T.; Naseer, A.; Stergioulas, L.K.; Young, T. (2010) Simulation in manufacturing and business: A review. *European Journal of Operational Research*, 203:1-13.

Law, A. *Simulation modeling and analysis*. Boston: McGraw-Hill Education, 2015.

Olsen, M.; Raunak, M. (2015) A method for quantified confidence of DEVS validation. In: *Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, 135-142.

Lopez-Paz, D., Oquab, M. Revisiting classifier two-sample test (2017). In: *International Conference on Learning Representations, Proceedings...* Toulon, France, pp. 1-15.

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.

Popovics, G., Pfeiffer, A.; Monostori, L. (2016) Generic data structure and validation methodology for simulation of manufacturing systems. *International Journal of Computer Integrated Manufacturing*, 29(12):1272–1286.

Raju, N.V.G.; Lakshmi, K.P.; Jain, V.M.; Kalidindi, A.; Padma, V. (2020). Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification. In: *Proceedings of the 2020 International Conference on Smart Systems and Inventive Technology*, pp. 729-735.

Sargent, R.G. (2013). Verification and validation of simulation models. *Journal of Simulation*, 7 (1):12-24.

Sargent, R.G.; Balci, O. (2017) History of verification and validation of simulation models. *In: Proceedings of the Winter Simulation Conference*, pp. 292–307.

Scheidegger, A.P.G.; Pereira, T.F.; de Oliveira, M.L.M.; Banerjee, A.; Montevechi, J.A.B. (2018) An introductory guide for hybrid simulation modelers on the primary simulation methods in industrial engineering identified through a systematic review of the literature. *Computers & Industrial Engineering*, 124:474-492.

Schriber, T.J.; Brunner, D.T.; Smith, J.S. (2017). Inside discrete-event simulation software? How it works and why it matters. *In: Winter Simulation Conference, Proceedings...*, pp. 735-749.

Schuirmann, D.J. (1987). A Comparison of the Two One-Sided Tests Procedure and the Power Approach for Assessing the Equivalence of Average Bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics*, 15(6):657-680.

Silva, G.T. da, Logan, B.R.; Klein, J.P. (2009). Methods for equivalence and noninferiority testing. Biology of blood and marrow transplantation, *Journal of the American Society for Blood and Marrow Transplantation*, 15(1):120-127.

Skoogh, A.; Johansson, B.; Stahre, J. (2012). Automated input data management: evaluation of a concept for reduced time consumption in discrete event simulation. *The Society for Modeling and Simulation International*, 88(11):1279-1293.

Sokolowski, J.A.; Banks, C.M. (2010). *Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains.* John Wiley and Sons.

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.

Thacker, B.H.; Doebling, S.W.; Hemez, F.M.; Anderson, M.C.; Pepin, J.E; Rodriguez, E.A. (2004). *Concepts of Model Verification and Validation*.

Tolosana, R.; Vera-Rodriguez, R.; Fierrez, J.; Morales, A.; Ortega-Garcia, J. (2020). Deep fakes and beyond: A Survey of face manipulation and fake detection. *Information Fusion*, 64:131-148.

Zeigler, B.P.; Nutaro, J.J. (2016) Towards a framework for more robust validation and verification of simulation models for systems of systems. *Journal of Defense Modeling and Simulation*, 13(1):3-16.

Yi, X., Walia, E., & Babyn, P. (2019). Generative adversarial network in medical imaging: A review. *Medical Image Analysis*, 58:1-24.