Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

USING GENERATIVE ADVERSARIAL NETWORKS TO VALIDATE DISCRETE EVENT SIMULATION MODELS

José Arnaldo Barra Montevechi
Gustavo Teodoro Gabriel
Afonso Teberga Campos
Carlos Henrique dos Santos
Fabiano Leal

Production Engineering and Management Institute
Federal University of Itajubá
Av. BPS, 1303
Itajubá, MG, 37500-903, BRAZIL

Michael E. F. H. S. Machado

Training Department
Flexsim Brazil
221 Regente Feijó
Campinas, SP, 13013-050, BRAZIL

ABSTRACT

Computer model validation is an essential step in simulation projects. The literature suggests using statistical techniques for comparing the outputs from the simulated model and the real system; however, statistical assumptions may be violated. Thus, Generative Adversarial 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 distributions with the same characteristics were submitted to verify the Power of the Test. The curves of each distribution were generated. In addition, a real case of a Discrete Event Simulation in a large emergency department was applied to the new validation technique. The results showed that GANs effectively discriminate data and can help validate computer models.

1 INTRODUCTION

Simulation models present tools, methods, and techniques to study complex systems. Moreover, in the simulation, the process can be improved by experimentation and optimization (Scheidegger et al. 2018). 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). However, the actions proposed in the model are only reliable and effective if the simulation model represents the real system (Sargent 2013). Computer model validation ensures it.

Montevechi et al. (2015) reviewed the main simulation methods, and, between eight frameworks, only one does not mention this step. Zeigler and Nutaro (2016) affirm that validation is the degree to which a model represents its system counterpart. Thacker et al. (2016) say that validation processes aim to quantify the model accuracy, comparing the simulation with experimental or operational results in the real world. In addition, the validation determines if the model represents the system for the desired purpose (Sargent and Balci 2017).

Although the step is essential for simulation projects, it is not an easy task to carry out. Validation may be used in real (can be measured) and hypothetical systems (cannot be measured). In the first case, the validation is carried out by comparing the results from the model built on the computer and the measurements performed on the physical system. In this approach, one should explore, as thoroughly as possible, the outputs of the real models with the simulated models. In hypothetical systems, the modeled

system is only created for a design, and it is generally not possible to obtain a satisfactory degree of model confidence (Sargent 2013).

However, Sargent and Balci (2017) state that there are more than 75 validation techniques. The authors divide them into subjective and objective techniques. Subjective techniques depend on the decision maker's knowledge and judgment, and they are considered the only possibility when no objective approach can be applied (Wang 2013). They hardly ensure the results obtained in simulation experiments. On the other hand, objective approaches are preferred since it uses mathematical or statistical methods to compare simulated models with observable systems. Real data are required to perform statistical procedures and determine the computer model validation. According to Balci (1994) and Wang (2013), objective techniques provide unique evidence, but their application requires experience and deeper knowledge by modelers.

Since objective approaches are preferable, the literature provides statistical tests to compare the simulation and real systems data. Many validation approaches are performed using univariate statistics. Modelers should choose between parametric tests (1 sample-t, 2 sample-t, F-test) or non-parametric tests (1 Wilcoxon sample, Mann-Whitney test), depending on data availability. If the variables of interest are random, the tests assess if the mean or variance from the simulation and real system is equal (Sargent 2013).

If the model aims to evaluate more than one metric, it is necessary to validate all of them. Balci (1994) and Sargent et al. (2016) claim that multivariate techniques should be used if there is a correlation between the outputs. Therefore, simultaneous confidence intervals show how the variables behave as a whole in the model. Simultaneous confidence intervals, Hoteling's T^2 test, and MANOVA are the most common tests used in the literature.

The tests consider only if the mean or standard deviation is the same statistically. They assume that the two compared datasets follow assumptions, such as minimum sample size, normality, and correlation. Moreover, the tests do not evaluate if the sample data may differ from another in a range. Since the validation model is not a binary variable and represents a confidence range from zero to 100% (Olsen and Raunak), we can use a tolerance to affirm whether the model is validated.

Some techniques of Deep Learning (DL) may be used to overcome the issue. Generative Adversarial Networks (GANs) may be used to train data and discriminate data since they can generate and judge data that presents non-linear behaviors and dependence between the variables (Agnese et al. 2019).

They were developed by Goodfellow et al. (2014) and are characterized as models of DL. The GANs are two Artificial Neural Networks (ANN) where the Generator $G(x)$ generates synthetic data and samples to fool the discriminating network. The Discriminator $D(x)$ distinguishes if the generated data is false or true. After the proper training, the Generator starts to provide samples like the real ones. In this sense, the paper aims to use GANs to discriminate outputs between the simulation model and the real system. Moreover, after the dataset judgments, an Equivalence Test is performed to verify if the difference in the dataset classification is inside a tolerance established by the modeler. In addition, to demonstrate the applicability of the proposed approach, we test it in theoretical distributions and a real study object.

The rest of the paper is organized as follows: section 2 gives the background on this work (deep learning in simulation projects, GANs, and model validation). Section 3 presents the proposed. Section 4 shows the results and discussions. Finally, section 5 gives the conclusion and directions for future studies.

2 RELATED LITERATURE

2.1 Deep Learning and Simulation

According to Ferreira et al. (2020), Artificial Intelligence (AI) used with the simulation has stood out as a robust solution aiming for efficient decisions. The AI is a set of tools that reproduce human behavior using computational resources, and, in this context, we highlight the DL techniques. According to LeCun et al. (2015), DL is based on multiple layers capable of learning data features with various levels of abstraction.

Ferreira et al. (2020) highlight that the DL approach can be used by several algorithms, such as artificial neural networks, fuzzy models, reinforcement learning, cellular automata, meta-heuristics, and big data

analytics. Moreover, we observe DL applications in several areas, such as image and speech recognition, web search, fraud detection, email filtering, and financial risk modeling (Choudhary et al. 2022).

When considering the use of DL techniques integrated with simulation projects, it is important to highlight that technological developments have increasingly encouraged this approach (De la Fluente et al. 2018). In this case, several applications stand out, such as in Civil Construction (Karim and Kim, 2020), Manufacturing (Wu et al. 2020), Services (De la Fluente et al. 2018), and IT (Nascimento et al. 2019).

The use of DL in simulation models allows the development of intelligent models integrated and capable of providing faster and most efficient decisions (Rodič, 2017; Santos et al. 2021). Moreover, Brailsford et al. (2014) highlighted the Discrete Event Simulation (DES) as the most popular operational research simulation technique and the most used in practice. Then, we have models that represent systems that evolve instantly at separate points in time (Law, 2014).

We highlight several applications involving the integration of DL and DES. Some works adopt DL techniques to solve complex problems and use DES to validate the proposed solutions (Nascimento et al. 2019; Wang et al. 2021; Karim and Kin, 2020; Wu et al. 2020). On the other hand, there are also studies where DL is used as an auxiliary technique aiming to promote more accurate inputs or even to optimize the experiments of DES models, such as the works proposed by De la Fluente et al. (2018) and Shi et al. (2020).

From our best knowledge, the only study that uses GANs and DES is Montevechi et al. (2021) that trained data as input in the data collection phase. Therefore, the present study is the first that uses GANs to validate DES models.

2.2 Model Validation in DES

Although the paper focuses on computer model validation, Balci (2010) states that validation is essential throughout all stages of DES projects. Thereby, it must occur cyclically (Popovics et al. 2016), continuously (Balci 2010; Wang 2013), and iteratively (Tsioptsias et al. 2016) over the model development (Foures et al. 2013). Part of the model must be built, verified, and validated before proceeding. Banks and Chwif (2011) suggest that models should be built from the simplest to the most complex details, facilitating validation and avoiding rework (reduces modeling time) when models become sophisticated, and errors are more difficult to find. Furthermore, when errors are detected earlier, better project quality is ensured (Balci 2010; Foures et al. 2013). Several models are carried out before reaching the final version since problems may appear throughout its construction (Sargent 2013).

Two or three groups of people should be involved in the computer validation stage: the model development team, the user of the simulation model, and/or an independent team. The first two groups are mandatory. When validation occurs by the modelers' team, the team itself decides the validity through evaluations and tests (self-validation). On the other hand, when the simulation user validates the model (co-validation), it should present a synergy with the team of modelers. Moreover, it should determine how satisfactory the model is at each stage of its construction (Kapoor and Shah 2016; Sargent 2013).

Validation may be used in real and hypothetical systems. The literature presents a set of techniques to help in validation, such as sensitivity analysis; animation; graphical analysis (histograms, boxplots, scatter plots); statistical tests (t-tests, confidence intervals, mean analysis, standard deviation, and variance); face-to-face validity and Turing test. Although there are many techniques, Sargent (2013) argues that mathematical or statistical methods are preferable because they provide objective decisions.

Finally, Balci (2010) notes that validity and acceptance range must be under the model's purpose. In this sense, Sargent et al. (2015) and Sargent et al. (2016) ensure that a range specifies the precision required in a simulation model. The range corresponds to the difference between the output variables of the real system and the simulated system with the upper and lower limit.

2.3 Generative Adversarial Networks

We can describe the Generative Adversarial Networks (GANs) as an AI technique, proposed by Goodfellow et al. (2014), focusing on generating synthetic (and realistic) data. Since it was proposed, the GANs have

been used in several fields, standing out in the health area (Yi 2019) and computer vision applications (Sorin et al. 2020). The GANs are based on two adversarial Artificial Neural Networks (ANN) that are trained iteratively and compete against each other. The two networks are the data Generator (G) and the data Discriminator (D) (Pan et al. 2019).

The generator G is a differentiable ANN (of parameters θ_g) that can generate synthetic samples (p_g) by mapping a latent input variable z (a noise with no practical meaning) to the real data space $G(z,\theta_g)$. Moreover, the discriminator $D(x,\theta_d)$ 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).

The discriminator D is trained based on the real data and the synthetic data generated by G. In the process, the objective is to maximize the probability of classifying both synthetic and real observations correctly. On the other hand, G is trained in order to minimize the probability of D identifying the synthetic data, minimizing $log(1-D(G(z, \theta_g),\theta_d)$.

After some iterations, we expect G to produce synthetic samples similar to the real ones, and consequently, the discriminator D will find it more difficult to classify them (Brownlee, 2020). Finally, if D does not differentiate the data, we conclude that the G achieved its goal, that is, it learned the real data distribution (Pan et al. 2019).

According to Goodfellow et al. (2014) and Pan et al. (2019), after the GANs are trained, the discriminator D will not be able to differentiate the real and synthetic data. Then, it will evaluate as a random probability, with a 50% chance of classifying the data correctly. However, since there is a double set of networks, achieving this level of precision is not an easy task (Bronwlee 2020).

Finally, although GANs are widely focused on images, videos, and sound processing, it is important to highlight that they can learn complex distributions that represent the behavior of a population, generating new samples with the same fit (Goodfellow et al. 2014). Therefore, we conclude that GANs can be used for the proposed approach, as described in the following section.

3 PROPOSED APPROACH

Montevechi et al. (2021) used GANs to generate synthetic data in the input data phase. Since the study tested the algorithm for four distributions and a real case and proved it efficient, we used the same algorithm to test the validation approach. 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 Phase (gray) and the Testing Phase (white), as shown in Figure 1. All steps were coded in Python using TensorFlow, Keras, and scikit-learn libraries.

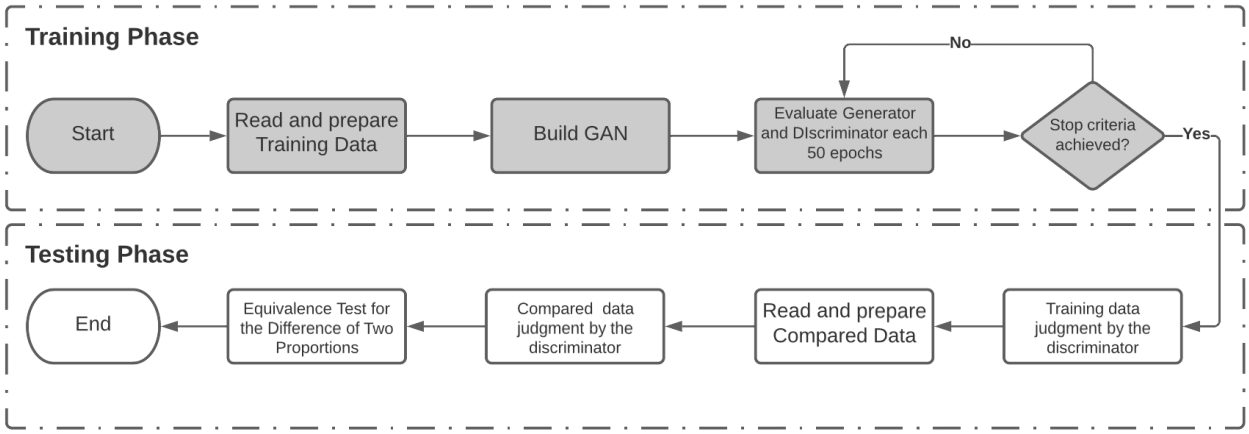

Figure 1: Framework to validation using GANs.

Once the algorithm wants to compare Real Data to Simulated Data (SD), the user must choose which one will be used in both phases. Montevechi et al. (2021) mentioned that data must be structured in tabular datasets, and each observation must be in a row while the attributes are in the columns. Attributes are considered the entities' characteristics, process cycle times, idleness, etc. In the second step, the algorithm trains the GANs. The discriminator access batches of synthetic and real data of equal size and try to classify the observations correctly. 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 the epoch is 10000. We used the same parameters in the algorithm that Montevechi et al. (2021) used in their study.

After completing the epoch training and every 50 epochs, the algorithm is evaluated. The evaluation guarantees that the synthetic samples correspond to the Training Data (TD). In this sense, the study uses the k-NN-based Classifier Two-Sample Tests (C2ST) (David and Oquab 2017, Cai et al. 2019) to classify the data. According to Cover and Hart (1967), in k-NN, the class is determined by observing the k nearest neighbors. Then, the classifier tries to separate the dataset with synthetic and training data, both scaled and in the same proportions. If the synthetic data are perfectly realistic, the k-NN algorithm classifies 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 AC reaches 100% (David and Oquab 2017). 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 only stop training if another condition is reached. The Discriminator evaluates the TD randomly if the Generator can trick the Discriminator (Brownlee 2020). The condition happens if the Discriminator discriminates the data by around 50.0%. 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%, because the judgment can assume a tolerance.

When the stop criteria are achieved, the Discriminator judge the TD. Then, the Compared Data (CD) are inserted into the program to be prepared. In this step, the data need to be the same way the judgment data was inserted. As soon as the data are inserted, they are rescaled based on the trained data, and then the Discriminator judges them.

An Equivalence Test for the Difference of Two Proportions (ETDTP) is carried out in the final step, where the proportion of two populations is given by *p1* and *p2*. The step comprises verifying statistically if the trained and compared data are equal. 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 (δ). The ETDTP measures if 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) hypotheses:

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

4 RESULTS AND DISCUSSION

4.1 Training Phase

We trained five statistical distributions following the training steps presented in section 3, as shown in Table 1. The training phase was carried out with four different sample sizes: 10000, 5000, 2000, and 1000. The training was performed for four amounts of data in the sample size. It measures how many epochs are necessary to achieve the conditions and the effect in the Power of the Test. Moreover, we changed the seed in the algorithm five times to verify how the judgment phase would behave.

The minimal level of accuracy desired is 95.0%. In this sense, after the training, the lower accuracy reached, on average, was by a bivariate normal (negative correlation) with 10000 data (96.62%). The higher accuracy was obtained by a bivariate normal (positive correlation) with 5000 data (99.34%). There is a pattern in training. In general, if the training phase presents more data, it is used to reach the conditions in less epoch than training with fewer input data quantities. Table 2 shows the accuracy and the number of epochs necessary to achieve the stop condition and the confidence interval for each metric.

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]]	$x_1x_2 = 0.9$ $x_1x_3 = -0.7$ $x_2x_3 = -0.6$

Table 2: Distributions used for train and power test.

Input Data	10000	5000	2000	1000
Normal Distribution				
Epoch	150.00 (47.22 - 252.78)	250.00 (66.66 - 433.34)	960.00 (232.30 - 1687.70)	910.00 (378.28 - 1441.72)
Accuracy	97.66% (97.00 - 98.32)	97.08% (96.53 - 97.63)	98.22% (97.30 - 99.14)	98.22% (97.34 - 99.10)
Bivariate Normal (positive correlation)				
Epoch	240.00 (55.62 - 424.38)	320.00 (219.10 - 420.90)	460.00 (338.39 - 581.61)	1060.00 (870.48 - 1249.52)
Accuracy	96.62% (96.18 - 97.06)	97.12% (96.50 - 97.74)	97.96% (96.84 - 99.08)	99.14% (98.48 - 99.80)
Bivariate Normal (negative correlation)				
Epoch	250.00 (147.22 - 352.78)	340.00 (255.70 - 424.30)	480.00 (284.99 - 675.01)	1180.00 (668.34 - 1691.66)
Accuracy	97.00% (95.51 - 98.49)	97.86% (97.35 - 98.37)	98.20% (97.71 - 98.69)	99.34% (98.66 - 100.00)
Bivariate Normal (no correlation)				
Epoch	240.00 (155.70 - 324.30)	270.00 (230.80 - 309.20)	390.00 (353.33 - 675.01)	670.00 (620.03 - 719.97)
Accuracy	97.22% (96.55 - 97.89)	97.74% (96.59 - 98.89)	97.84% (96.88 - 98.80)	99.06% (97.93 - 100.00)
Multivariate Normal				
Epoch	250.00 (180.70 - 319.30)	340.00 (282.86 - 397.14)	500.00 (287.54 - 712.46)	670.00 (501.40 - 838.60)
Accuracy	97.58% (96.44 - 98.72)	97.96% (96.83 – 99.09)	98.16% (97.36 - 96.83)	98.50% (98.36 - 98.64)

4.2 Power of the Test

The Power of a statistic test evaluates the probability of rejecting H_0 if H_0 is false (Montgomery and Runger 2019). On the other hand, the value β (Type II error) is the difference between a 100% chance of getting the correct answer and the Power. Therefore, in ETDTP, the Power of the Test is the probability of ensuring that the difference between the two sample proportions is within the equivalence limit.

Since the statistical tests are subject to errors and show how assertive the proposed approach is, we carried out some tests with the Discriminator for each trained distribution. First, we trained the normal distribution with a sample size of 10000. After reaching the stop conditions ($A_C \geq 95.0\%$ and $45.0\% \leq$ Discriminator $\leq 55.0\%$), other normal distribution with the same parameters as the trained was generated (CD) with a sample size of 10000, and the ETDTP was performed. Since both distributions have the same parameters, we expect that the Discriminator judges both datasets statically with a tolerance range. In other words, we check whether the proposed method can distinguish data from two statistical distributions with the same parameters. It is expected that it cannot distinguish, validating the approach.

The steps of generating a new normal distribution (same parameters as the trained data set), the Discriminator judge the new dataset and performs the ETDTP considering a tolerance of 5.0% was performed 10000 times. Again, we repeat the same procedure, changing the tolerance (δ) to 10.0%. Moreover, we varied the number of data submitted in the second stage to 5000, 2000, 1000, 500, 100, and 10. The process was repeated five times to obtain a confidence interval for the Power Test.

After performing all procedures mentioned above, we also changed the amount of data in the training phase. The normal distribution was trained with 5000, 2000 and 1000 data. All the five distributions were trained, and the Discriminator judged them. Figure 2 shows the curves generate for each distribution, considering the tolerance of 5.0% for distributions (a), (b), (c) and (e) and 10.0% for distributions (a) and (d).

According to the curve (1.a), with 10000 data in training and compared data, the Discriminator presented an assertiveness of 97.09%, considering 5.0% of tolerance. If we consider 5000 data in the second phase, the Power decrease to 83.35%. The algorithm did not detect similarity when 1000 data were tested in the Training Phase. According to Cohen (1988) and Brydes (2019), power is conventionally set as 80%. Therefore, we considered it necessary at least 80.00% success to have an efficient Power.

On the other hand, the curve for 10.0% of tolerance (1.b) showed no difference between the results presented by samples with 10000 and 5000 in the Training Phase. When the Discriminator judged normal distributions presenting 10000 and 5000 samples in the first phase, both got 100.0% of success. However, if the sample size in the compared dataset decreases to 1000, the percentage of success also decreases to 81.30%. For normal distributions with a sample size of 500, the Power of the Test stays around 38.72%. The curves with 2000 and 1000 samples in the Training Phase showed the same behavior.

The three normal bivariate distributions present almost the same results considering the number of epochs to achieve the stop conditions, accuracy, and the Discriminator judgments. More data is requested for distributions with more than one dimension in the Training and Testing phase. The Power for 1000 sample size (TD) did not achieve at least 80.00% in any case. The maximum percentage of success was 44.44%, judging Bivariate distribution with negative correlation and 10000 data in the Testing Phase and 10.0% of tolerance. The same results are presented in the Bivariate distributions trained with 2000 data.

The distributions with no correlation got at least 80.00% of correct judgments if we consider 10000 data in the first phase and at least 1000 in the second for a tolerance of 10.0%. According to curve (4.b), we recommend a sample size of 2000 in the TD and at least 5000 in the CD. However, the same or better results can be achieved if we use at least 5000 data in the first dataset and 2000 in the second one.

By increasing one dimension in the distributions (Multivariate Distribution), the number of epochs needed to achieve the stop conditions also increased. We concluded that the more dimensions it wants to test at once, the more data it is requested. The results for the judgment considering 1000 data in the first phase showed that a smaller sample size makes the Power drop off. In conclusion, the algorithm needs more

epochs to provide reliable results if we provide fewer data in the training phase. Moreover, testing more than one variable for each observation needs a higher samples size to provide at least 80.0% of the Power.

In conclusion, we suggest using the dataset that presents more data in the Training Phase. The more data the algorithm receives in the Training phase, the better the judgment of the data and the smaller the confidence interval of the Power curves. Since the approach aims to test if the outputs from the real model are equal to the real system, we suggest using the simulated data in the first phase. Simulated data is easier to generate by the software than collected in the real system and can be imputed in the training of the GANs.

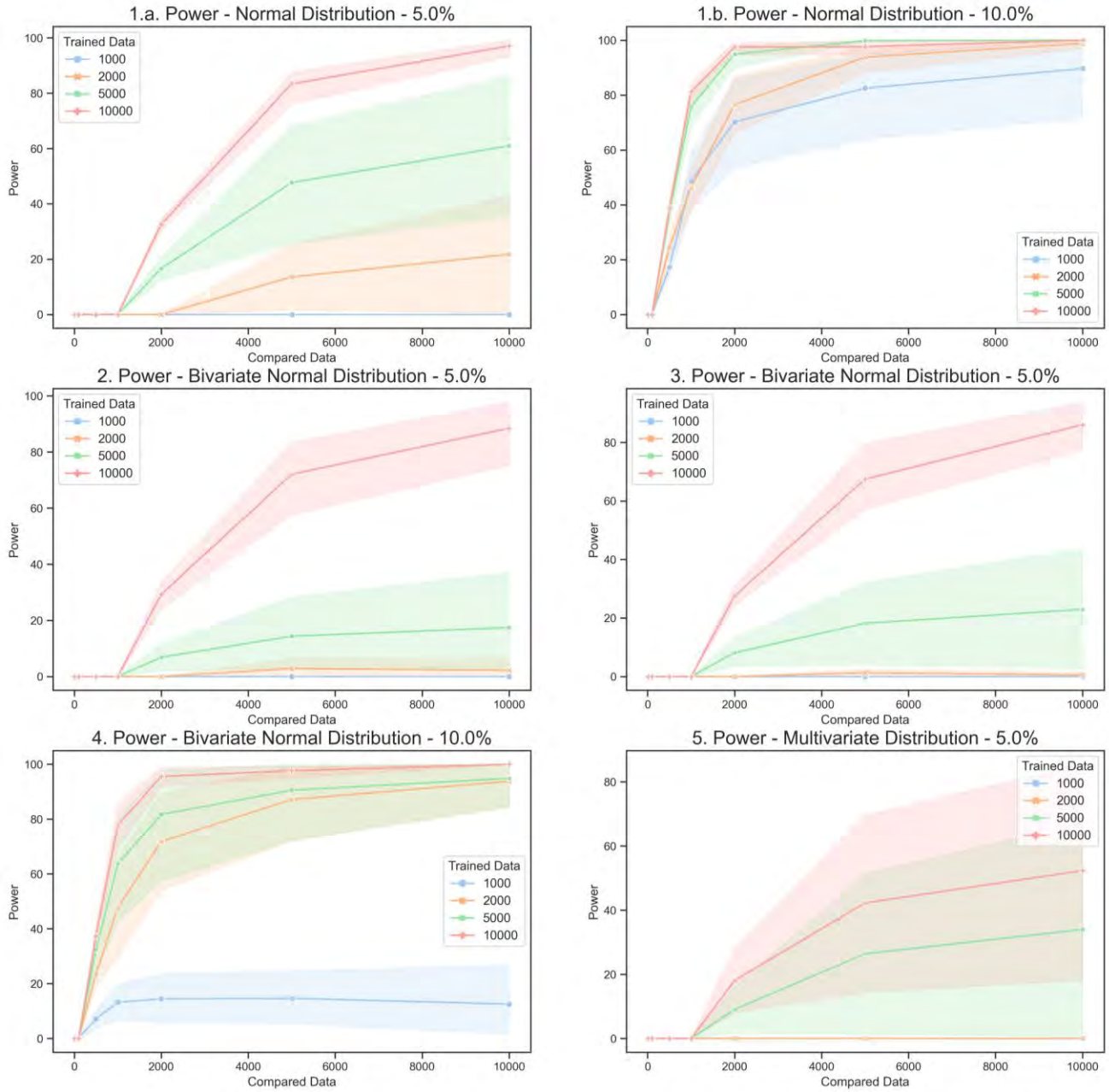

Figure 2: Curves for the Power of the Test.

4.3 Case Study – Emergency Department

4.3.1 Patient flow

We applied the technique in a large emergency department (ED) in Brazil to test the approach. The (ED) serves around 400 people/day and has a variety of flows, where the patient can make appointments, tests (blood, urine), diagnostic imaging tests (X-ray, ultrasound, etc.), surgeries, and hospitalization.

Patients arrive at the ED, and they get a password and wait to be triaged. In triage, the patient is classified into five colors: red, orange, yellow, green, and blue. Upon being diagnosed with an orthopedic procedure, the patient receives a medical evaluation, and its registration is made at the same time as the evaluation by a companion. After the evaluation, depending on the severity, he needs surgery (doctors and nurses) or just an orthopedic procedure, which an orthopedic technician performs. If the patient is considered an emergency, he goes to the emergency room. Finally, if the patient is classified as mild, he/she awaits his/her registration and the medical evaluation.

After this point, the flow is the same for the three types of patients. The patient performs up to five procedures that can occur in parallel or when the resources and locations are available. The five procedures are: waiting for the specialist, stabilization, medication, waiting for hospitalization, or hospitalization.

If the patient needs to wait for a doctor, he waits for him and then goes through the appointment. If he needs the stabilization, he is stabilized. However, if a medication is requested, the nursing technician performs it. The patient may receive a second medication, take a collection for examination, perform an X-ray, ultrasound, tomography, electrocardiogram, echocardiogram or wait for an external procedure. If the patient needs to carry out more than one of the procedures, it is done as soon as the resources are released. If he needs hospitalization, he waits for his release, and then he is hospitalized. If hospitalization occurs before the other procedures, the exams, collection, medication, and appointments are carried out in the bed. Finally, the patient is released and leaves the system. The model was built in the FlexSim® software.

4.3.2 Model Validation

We validated the model using three metrics: door-to-triage time (DTT), door-to-doctor time (DTD), and Length of Stay (LOS) of patients classified as green. The data from the real system were collected through the hospital's system, which controls the times through records time made by the team. For the validation and the input in the GANs, we used the Simulated Data for training and the Real Data for the Test Phase. Data were collected from 6684 patients from the real system, while the simulation provided 12456 data. It was necessary 600 epochs to achieve the stop conditions. The model got a final *Ac* equivalent to 96.6% of the imputed data. The GANs discriminator judged the TD at 48.67%, while the CD were classified by 34.96%.

According to the results presented by the Equivalence test (δ = 5.0%), it was not possible to affirm that the difference in the classification of the real system and the simulated model is within the tolerance range (p-value = 1.000). The Power of the Test was 0.0%, indicating that the probability of committing the Type II error is 100.0%. Despite this, the model can be validated with a minimum tolerance of 29.84% (p-value: 0.049) estimated between the datasets.

We highlight that the proposed approach compares the data considering the data distribution, mean, and standard deviation. In this sense, the confidence of the test is 95.0%. If the validation is carried out with tests already present in the literature, it is necessary to compare the data by means and standard deviation. Liu (2022) says if more than one test is performed together, the confidence is calculated by $(1-\alpha)^K$, where K is the number of tests. Therefore, if the validation is done individually, the model confidence drops to 73.51% (0.95^6) because six statistical tests are needed (3 to evaluate means and 3 to evaluate deviations).

For comparison, we performed the validation of each metric separately. The same amount of data was imputed in each training of the three metrics. We kept the sample size in both datasets. For the DTT, the GANs could not reach a minimum final *Ac* of 95.0% considering 10000 epochs. However, it was possible

to reach a final *Ac* equivalent to 93.6% in 2300 epochs. The Discriminator judged the TD equal to 53.59% and the CD at 52.72%. After the Equivalence test, we can affirm that the difference between the data classification data is statistically within the tolerance of 5.0% (p-value = 0.029). The Power of the Test is 100.0%, indicating that the probability β is 0.0%.

Regarding the DTD, it took 300 epochs to reach a final *Ac* of 95.7%. TD were judged at 49.12%, while CD scored 50.18%. The difference between the classification of the two datasets is statistically within tolerance (p-value = 0.029) with a Power Test of 99.9%.

Finally, 700 epochs were needed for LOS validation with a final *Ac* of 98.4%. The Discriminator rated the first phase data at 45.69% and the second phase data at 38.44%. The results show that it is not possible to affirm that the difference in the classification of the DS and the real data is within the tolerance interval (p-value = 1.000). The probability of making the Type II error is 100%, while the Test Power is 0.0%. Since the LOS cannot be validated individually considering a real tolerance of 5.0%, the tests were performed, and the minimum required tolerance is 16.96% (p-value: 0.4961). Therefore, their confidence drops to 85.74% (0.95^3). Table 3 shows the summary of the validation with the new approach and 2 sample-t and or Hotelling T^2 test.

Table 3: Validation summary.

Metric	Epoch	Ac	TD	CD	Validation (5.0%)
DTT, DTD, LOS	600	96.6%	12456	6684	29.84%
DTT	2300	93.6%	12456	6684	✓
DTD	300	95.7%	12456	6684	✓
LOS	700	98.4%	12456	6684	16.96%

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. The first one comprises generating the data until the artificial networks achieve the necessary accuracy defined by the modeler, and the Discriminator judges the data between 45.0% and 55.0%. The second phase comprises submitting the training data to be judged and after it is performed the Test for Difference of Two Proportions.

We carried out tests with the Discriminator for a normal distribution, a bivariate normal distribution with positive, negative, and no correlation, and a normal multivariate. All of them were trained with a sample of 10000 in the TD, 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, which had to indicate that the TD and CD were statistically significant. The process was repeated with the same distribution, however, with 5000, 2000, 1000, 100, and 10 data. Moreover, we changed the sample size in the first phase, and all procedures were repeated five times. The curves for each distribution for 5.0% and 10% tolerance were generated. Finally, we applied the new approach in a large ED, and the validation was performed considering three outputs.

The results showed that GANs efficiently discriminate data and can help in the model validation phase in DES. However, more than 5000 data in both phases is necessary to achieve reliable results. For future works, we suggest training more distributions, such as discrete and conditional variables. Moreover, we suggest training the distributions using the variants of GANs, such as cGAN, DCGAN, and WGAN.

ACKNOWLEDGMENTS

The authors would like to express their gratitude to CNPq, CAPES, and FAPEMIG for their support throughout this research. Moreover, the authors would like to thank FlexSim Brasil for the support and knowledge provided in the study.

REFERENCES

Agnese, J., J., Herrera, H. Tao, X. Zhu. 2020. "A survey and taxonomy of adversarial neural networks for text-to-image synthesis". *WIREs Data Mining Knowledge Discovery* 1: 1–26.

Balci, O. 1994. "Validation, verification, and testing techniques throughout the life cycle of a simulation study". *Annals of Operations Research*, 53, 121-173.

Balci, O. 2010. "Golden rules of verification, validation, testing, and certification of modeling and simulation applications". *SCS M&S Magazine*, 1(4): 1-7.

Banks, J., L. Chwif. 2011. "Warnings about simulation". *Journal of Simulation*, 5(4) 279-291.

Brailsford, Sally, Leonid Churilov, and Brian Dangerfield. 2014. *Discrete-event simulation and system dynamics for management decision making.* 1. Ed. John Wiley & Sons

Brownlee, J. 2020. *Generative Adversarial Networks with Python: Deep Learning Generative Models for Image Synthesis and Image*. 1. ed. Machine Learning Mastery.

Brydges C. R. 2019. "Effect size guidelines, sample size calculations, and statistical power in gerontology". *Innov Aging*, 3(4) 1-4.

Cai, L, Y., Chen, N. Cai, W. Cheng, H. Wang. 2020. "Utilizing Amari-Alpha divergence to stabilize the training of generative adversarial networks". *Entropy* 22: 1-19.

Choudhary, K., B. DeCost, C. Chen, A. Jain, F. Tavazza, R. Cohn, C. W. Park, A. Choudhary, A. Agrawal, S. J. L. Billinge, E. Holm, S. P. Ong, C. Wolverton. 2022. "Recent advances and applications of deep learning methods in materials science". *Compuational Materials*, 8(59): 1-26.

Cohen, J. 1988. *Statistical Power Analysis for the Behavioral Science*. 2 ed. Hillside, NJ: Lawrence Erlbaum Associates.

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

David, L. P., M. Oquab. 2017. "Revisiting classifier two-sample tests". *In Proceedings of the International Conference on Learning Representations*, 1-15.

De La Fluente, R., I. Erazo, R. L. Smith. 2018. "Enabling intelligence processes in simulation utilizing the tensorflow deep learning resources". In *Proceedings of the Winter Simulation Conference*, 1108-1119.

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

Ferreira, W. P, F. Armellini, F, L. A. Santa-Eulalia. 2020. "Simulation in industry 4.0: a state-of-the-art review". *Computers & Industrial Engineering* 149: 1-17.

Foures, D., V. Albert, A. Nketsa. 2013. Simulation validation using the compatibility between simulation model and experimental frame. In *Proceedings of the Summer Computer Simulation Conference*, 326-332.

Goodfellow, I. J, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. 2014. "Generative adversarial nets". *arXiv*, 1: 1-9.

Kapoor, R., B. J. Shah. 2016. "Simulation model for closed loop repairable parts inventory system with service level performance measures". *International Journal of Services and Operations Management*, 23:18-42.

Karim, M. M., C. H. Dagli, R. Qin. 2020. "Modeling and simulation of a robotic bridge inspection system". *Procedia Computer Science*, 168: 177-185.

Law, A. M. 2014. *Simulation Modeling and Analysis*. 5. ed. Boston: McGraw-Hill Science.

Lecun, Y., Y. Bengio, G. Hinton. 2015. "Deep Learning". *Nature*, 521: 436-444.

Liu, R. 2022. "Statistical guideline #7 adjust type 1 error in multiple testing". *International Journal of Behavioral Medicine*, 29:137–140.

Montevechi, J. A. B., A. T. Campos, G. T. Gabriel, C. H. dos Santos. 2021. "Input data modeling: an approach using generative adversarial networks". In *Proceedings of the Winter Simulation Conference*, 1-12.

Montevechi, J. A. B, Pereira, T. F; Silva, C. E. S.; Miranda, R. C.; Scheidegger, A. P. G. 2015. "Identification of the main methods used in simulation projects". In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W K. V. Chan, 1 Moon, T. M K. Roeder, C. Macal, and M D. Rossetti, 3469-3480. Huntington Beach, California.

Montgomery, D. C., G. C. Runger. 2018. *Applied Statistics and Probability for Engineers*. 7 ed. Wiley.

Nascimento, I., R. Souza, S. Lins, A. Silva, A. Klautau. 2019. "Deep reinforcement learning applied to congestion control in fronthaul networks". In proceedings of the IEEE Latin-American Conference on Communications, 1-6.

Pan, Z., W. Yu, X. Yi, A. Khan, F. Yuan, Y. Zheng. 2019. "Recent progress on generative adversarial networks (GANs): a survey". *IEEE Access* 7: 36322-36333.

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

Raju, N.; Lakshmi, K.; Scholar, V.; Kalidindi, A.; Padma, V. 2020. "Study of the influence of normalization/transformation process on the accuracy of supervised classification". In *Proceedings of the Third International Conference on Smart Systems and Inventive Technology*, 729-735. Online.

Rodič, B. 2017. Industry 4.0 and the New Simulation Modelling Paradigm. *Organizacija* 50(3): 193-207.

Santos, C. H., J. A. B. Montevechi, J. A. de Queiroz, R. C. de Miranda, F. Leal. 2021. "Decision support in productive processes through DES and ABS in the digital twin era: a systematic literature review". *International Journal of Production Research*, 1-21.

Sargent, R. G. 2013. Verification and Validation of Simulation Models. *Journal of Simulation* 7(1): 12–24.

Sargent, R. G. 2015. "An interval statistical procedure for use in validation of simulation models". *Journal of Simulation*, 9 (3): 232-237.

Sargent, R. G., D. M. Goldsman, T. Yaacoub. 2016. "A tutorial on the operational validation of simulation models". In *Proceedings of the Winter Simulation Conference*, 163-177.

Sargent, R. G., O. Balci. 2017. "History of verification and validation of simulation models". In *proceedings of the Winter Simulation Conference*, 292-307.

Scheidegger, Anna Paula. Galvao, T. F. Pereira, M. L. M. Oliveira, A. Banerjee, J. A. B. Montevechi. 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.

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.

Shi, D., W. Fan, Y. Xiao, T. Lin, C. XING. 2020. "Intelligent scheduling of discrete automated production line via deep reinforcement learning". *International Journal of Production Research*, 58(11), p. 3362-3380, 2020.

Sokolowski, J. A, C. M. Banks, C. M. 2010. *Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains*. 1 ed. New Jersey: John Wiley & Sons.

Sorin, V., Y. Barash, E. Konen, E. Klang. 2020. "Creating artificial images for radiology applications using generative adversarial networks (GANs) – a systematic review". *Academic Radiology* 27(8): 1175-1185.

Tsioptsias, N., A. Tako, S. Robinson. 2016. Model validation and testing in simulation: a literature review. In *Proceedings of the 5th Student Conference on Operational Research*, 1-11.

Wang, S., X. Gong, M. Song, C. Y. Fei, S. Quaadgras, J. Peng, P. Zou, J. Chen, W. Zhang, R. J. Jiao. 2021. "Smart dispatching and optimal elevator group control through real-time occupancy-aware deep learning of usage patterns". *Advanced Engineering Informatics*, 48: 1-16.

Wang, Z. Selecting verification and validation techniques for simulation projects: a planning and tailoring strategy. In *Proceeding of the Winter Simulation Conference*, 1233-1244.

Wu, C. H., F. Y. Zhou, C. H. Tsai, C. J. Yu, S. Dauzère-pérès. 2020. "A deep learning approach for the dynamic dispatching of unreliable machines in re-entrant production systems". *International Journal of Production Research*, 58 (9), 2822-2840.

Yi, X., E. Walia, P. Babyn. 2019. "Generative adversarial network in medical imaging: a review". *Medical Image Analysis*, 58: 1-20.

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

AUTHOR BIOGRAPHIES

JOSE ARNALDO BARRA MONTEVECHI is a Titular Professor of the Production Engineering and Management Institute at the Federal University of Itajubá, in Brazil. He holds the degrees of Mechanical Engineer from the Federal University of Itajubá, M.Sc. in Mechanical Engineer from the Federal University of Santa Catarina, and Doctorate of Engineering from Polytechnic School of the University of São Paulo. His research interest includes Operational Research, Simulation, and Economic Engineering. His email address is montevechi@unifei.edu.br.

GUSTAVO TEODORO GABRIEL is a Ph.D. student in Industrial Engineering at the University of Itajubá in Brazil. He holds his bachelor's and master's degrees in Industrial Engineering from the Federal University of Itajubá. His research areas include Process Mapping, Simulation, Validation, and Machine Learning. His email address is gustavo.teodoro.gabriel@gmail.com.

AFONSO TEBERGA CAMPOS is a Ph.D. student in Industrial Engineering at the University of Itajubá in Brazil. He holds his bachelor's and master's degrees in Industrial Engineering from the Federal University of Itajubá. His research areas include Simulation, Artificial Intelligence, and Machine Learning. His email address is afonso.teberga@gmail.com.

CARLOS HENRIQUE DOS SANTOS is a Ph.D. Student in Industrial Engineering at the Federal University of Itajubá, in Brazil. His bachelor's and master's degrees in Industrial Engineering from the Federal University of Itajubá. His research interest includes Simulation, Industry 4.0, Digital Twins, and Simulation-based optimization. His email address is chenrique.santos@unifei.edu.br.

FABIANO LEAL is a Professor in Industrial Engineering at the Federal University of Itajubá in Brazil. He holds the degrees of Mechanical Engineer from Federal University of Itajubá and M.Sc. in Industrial Engineering from Federal University of Itajubá, and Doctorate of Mechanical Engineer from Universidade Estadual Paulista. His research interest includes Business Process Modeling, Discrete Event Simulation, Design, and Measurement of Work. His email address is fleal@unifei.edu.br.

MICHAEL E. F. H. S. MACHADO is the Business Manager at Flexsim Software Corporate Brazil. He has been working with Discrete Event Simulation since 2007, leading Brazilian Flexsim Software operations and business. He holds a Master and Certificater in Bussines Adminstration at Insper and IBMEC. His research interests include leadership and simulation, focusing in optimization. His email address is michael.machado@flexsimbrasil.com.br.