*Proceedings of the 2021 Winter Simulation Conference*
*S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo and M. Loper, eds.*

# INPUT DATA MODELING: AN APPROACH USING GENERATIVE ADVERSARIAL NETWORKS

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

Production Engineering and Management Institute
Federal University of Itajubá
Itajubá, MG, 37500-903, BRAZIL

## ABSTRACT

Input data modeling varies according to the modeler's objectives and may be a simple or complex task. Despite great advances in data collection techniques, the input data analysis remains a challenge, especially when the input data is complex and cannot be modeled by standard solutions offered by commercial simulation software. Therefore, this paper focuses on how Generative Adversarial Networks (GANs) may support input data modeling, especially when traditional approaches are insufficient or inefficient. We evaluate the adoption of GANs for modeling correlated data as well as independent and identically distributed data. As results, GAN input models were able to generate highly accurate synthetic samples (average accuracies > 97.0%). For univariate distributions, we found no significant difference between standard and GAN input models performances. On the other hand, for correlated data, GAN input models outperformed standard ones. The most relevant accuracy gain was observed for the bivariate normal.

## 1 INTRODUCTION

Computer simulation has stood out in recent decades as a powerful tool to support decisions. Law (2014) highlights its use in several areas, e.g., manufacturing, financial systems, transport, services, and military processes. In this area, Discrete Event Simulation (DES) stands out as the most used computer simulation technique (Scheidegger et al. 2018). It remains an outstanding tool, even considering the modern decision-making context, the evolution of systems and technologies, and the need for increasingly faster and more flexible decisions (Mourtzis 2020). This paper focuses on DES models and from now on it will be designated as 'simulation'.

Despite its great applicability, many decision makers have failed to take advantage of simulation. One of the main factors that contribute to this difficulty is the high time spent in simulation projects (Skoogh et al. 2012). Considering all necessary steps to develop a simulation project, Robertson and Perera (2002) highlight that the input data modeling is one of the most time-consuming stage, representing about 20% to 50% of the total time. Although Skoogh and Johansson (2009) state that the most critical activities are related to data collection, we observe that this activity has been facilitated by recent advancements of technologies and solutions. However, input data analysis remains a challenge, especially when input data are complex and cannot be modeled by standard solutions offered by commercial simulation software.

.

**978-1-6654-3311-2/21/$31.00 ©2021 IEEE**

Previous studies broadly address the use of parametric probability distributions to represent model inputs, e.g., process times, entities' arrival rates, and demands for products and services. However, as input data may be highly complex, fitting probability distributions may represent a challenging task (Kuhl et al. 2007). For example, available parametric distributions may not capture input data singularities, such as multimodalities. In addition, modelers and simulation software usually assume that data are independent and identically distributed, even though this assumption does not always hold true (Cen et al. 2019). As a result, simulation practitioners may develop models that are not able to represent real systems with the desired accuracy, hindering decision-making (Robertson and Perera 2002).

Over the years, studies proposed a range of statistical tools for modeling input data with specific characteristics, such as copulas for interdependent data and ARIMA for time series (Leemis 1997). However, it is important to highlight that these tools usually require specific knowledge, which limits their adoption. In addition, commercial simulation software and their input modeling functionalities do not support most of these techniques (Cen et al. 2019). To overcome such limitations, it is necessary to develop tools to support input data modeling for complex data. We highlight the potential use of Artificial Neural Networks (ANNs) or, more precisely, Generative Adversarial Networks (GANs). According to Agnese (2020), GANs comprise a powerful and flexible Artificial Intelligence (AI) technique, which can learn complex patterns. Moreover, Goodfellow et al. (2014) highlight that GANs enable representing complex probability distributions and generating new synthetic (but realistic) data. This approach is in consonance with the conclusion of Ferreira et al. (2020) who highlight the growing tendency to use AI techniques in simulation projects.

Therefore, considering the issues related to the input data modeling phase, this paper focuses on how GANs may support input data modeling, especially when traditional approaches are insufficient or inefficient. We evaluate the adoption of GANs for modeling correlated data as well as independent and identically distributed data. Moreover, to demonstrate the applicability of the proposed approach, we test it in theoretical cases and in a real study object. The rest of the paper is organized as follows: section 2 gives the background on this work (input data modeling in simulation projects and GANs). The proposed approach is described in section 3. Section 4 is dedicated to the results and discussions. Finally, section 5 presents conclusion and directions for future studies.

## 2 RELATED LITERATURE

### 2.1 Input data modeling

The main goal of input data modeling is to represent random inputs of the studied system, based on real data, capturing their main characteristics (Biller and Gunes 2010). Banks et al. (2010) state that input data modeling varies according to the modeler's objectives and may be a simple or complex task. Furthermore, input data modeling plays an important role in simulation projects since it affects its validation and subsequent decision making (Sargent 2013).

Input data modeling may represent a barrier to simulation adoption for its time consumption. Robinson (2004) reports that, depending on the research field, simulation projects may be planned and executed for months, while Bokrants et al. (2015) highlight that projects applied in the daily business need greater agility. In this case, considering studies in companies from the different sectors, Barlas and Heavey (2016) add that there is a need for changes and developments that allow faster and, consequently, cheaper simulation projects. As input data modeling usually consumes a considerable part of the time spent in simulation projects (Skoogh and Johansson 2009), we consider it as a promising research field.

Skoogh and Johansson (2008) highlight that the main activities considered in this stage include the identification, collection, modeling, and validation of input data models. Data identification consists of defining relevant parameters for the simulation model and the desired level of detail, while data collection

comprises the choice and use of an appropriate and available collection method (Skoogh and Johansson 2008). After that, in the modeling phase, data are converted into empirical or statistical representations, which portray the real behaviors. Biller and Gunes (2010) divided data modeling into three steps: (i) select one or more statistical distributions that fit the data; (ii) determine its parameters; and (iii) verification of the adjustment quality via tests and graphical analyzes. Finally, it is desirable to validate the input data model. According to Sargent (2013), there are not many procedures available for that purpose, and the most common is qualitative validation.

Data collection presents some recurring issues, such as limitations on data availability (Bokrants et al. 2015) and data quality (Robertson e Perera 2002). However, recent technological solutions may substantially mitigate them. Data collected by sensors, smart devices, and automated systems are becoming a reality, leading to faster and more reliable data collection (Skoogh et al. 2012, Barlas and Heavey 2016). In this sense, several approaches have emerged in recent years to make data collection more efficient, such as Data-driven Simulation (Sormaz and Malik 2018), Real-time Simulation (Saez et al. 2018), Simulation as a Cyber-physical System (Montevechi et al. 2020), and Online Simulation (Scholl et al. 2010).

On the other hand, data modeling still must overcome some challenges, especially regarding to data complexity (Kuhl et al. 2008, Cen et al. 2019). Data complexity directly affects simulation models' accuracy since its results depend on how precisely the input data model represents the real systems (Robertson e Perera 2002). Although several standard statistical distributions are available to modelers, there are cases where none of them is a good fit (Biller and Gunes 2010). The author state that, in these cases, DES practitioners may adopt empirical distributions for representing the input uncertainty. Nevertheless, simulation platforms usually support only univariate empirical distributions.

If the project presents input data with behaviors that cannot be represented by standard statistical distributions or even empirical ones, it is necessary to use techniques capable of capturing these behaviors, such as multivariate distributions, Markov chains and processes, ARMA and ARIMA models and homogeneous or non-homogeneous Poisson processes (Leemis 1997). However, as also stated by Rodič (2017), we recognize the need for friendly tools and models, which need little knowledge on the part of the decision-maker. Finally, although there are several data characteristics that require specific solutions for data modeling, in this study we only address the multivariate problem.

### 2.2 Generative Adversarial Networks (GANs)

Synthetic contents generated through deep learning are popularly defined as "deep fake". The term came up 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. In the following years, "deep fake" contents were used for other harmful purposes, such as fake news and financial fraud, causing considerable concern on authorities around the world (Tolosana et al. 2020).

However, there are also positive uses of synthetic contents, especially in the healthcare area. Sorin et al. (2020) review studies that aim to generate realistic medical imaging results. Since synthetic images are not real, they do not violate patient confidentiality and data protection requirements. 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. It would be possible to improve radiography image quality even reducing radiation doses. Thereby, these exams could be carried out more frequently without compromising patients' health.

GANs, an AI technique proposed by Goodfellow et al. (2014), can be used for generating "deep fake" content, i.e., synthetic data close to real data. This purpose is in line with that of input data modeling: to generate realistic (but usually not the same) input data. Based on game theory, GANs comprise two adversarial ANNs, 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 $\theta_g$) that learns

to generate synthetic samples $p_g$ by mapping a latent input variable $\mathbf{z}$ (a noise, with no practical meaning) to the real data space $G(\mathbf{z},\theta_g)$. $D(\mathbf{x},\theta_d)$ is also a differentiable ANN (of parameter $\theta_d$) which outputs a single scaler representing the probability that an observation $\boldsymbol{x}$ will originate from the real data and not from $p_g$ (Goodfellow et al. 2014).

According to the authors, D is trained with access to real data and synthetic observations generated by G. Therefore, its 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 $log\ (1 - D\ (G\ (\mathbf{z},\ \theta_g),\ \theta_d)$. Thus, the greater the probability resulting from D (which indicates that the observation is genuine) the closer to zero will be the logarithm value. For a more stable learning, GAN training is performed iteratively, with only one ANN learning at a time.

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). Thus, Pan et al. (2019) conclude that the generator was able to achieve its goal and learned the real data distribution. Figure 1 shows how the GANs' training is performed for image generation.

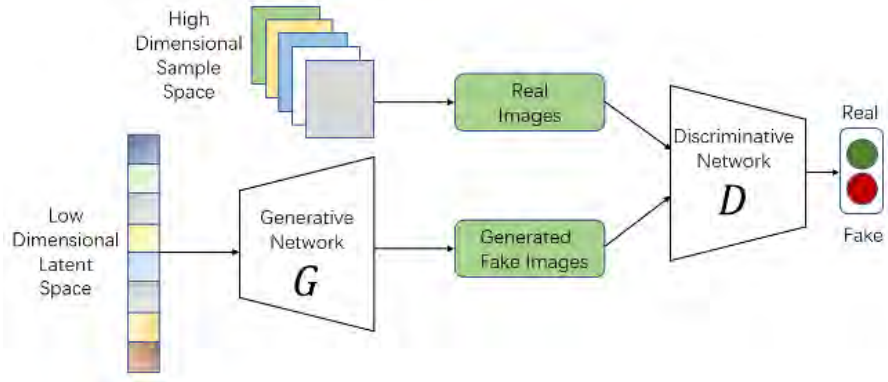


Figure 1: GANs' architecture. (Adapted from Cai et al. (2020))

Yi, Walia, and Babyn (2019) claim that GANs have been successful in several areas and have grown fast, being used to synthesize medical images and increase sample sizes (mitigating the problems of data shortages and overfitting). In general, GANs have positively impacted the computer vision area (Sorin et al. 2020). However, despite receiving more attention for image, video, and sound processing, GANs are not limited to these data. Goodfellow et al. (2014) argue that, regardless of the application, GANs are able to learn complex distributions that represent the behavior of a population and can generate new coherent samples with the same behavior.

## 3 PROPOSED APPROACH

Some steps should be carried out to build input models using GANs. We propose the following framework to guide this process, as presented in Figure 2.

Data reading means retrieving input data from the source database. The developed algorithm requires structured tabular datasets, such as worksheets. Each observation should be recorded as a row while columns should contain observation attributes. From a simulation perspective, observation attributes examples are process cycle times, inter arrival times, and entities' characteristics.

One advantage of using GANs for input data modeling is to be able to model joint distributions, i.e., probability distributions for two or more random variables (Du et al. 2021). In these situations, the joint distribution allows capturing relationships between input variables. For example, in healthcare environments there may be correlated patient cycle times due to individual patient characteristics, such as

age and health condition. To model a joint distribution, all observation attributes in a row must be related to the same observation, e.g., same item or patient.

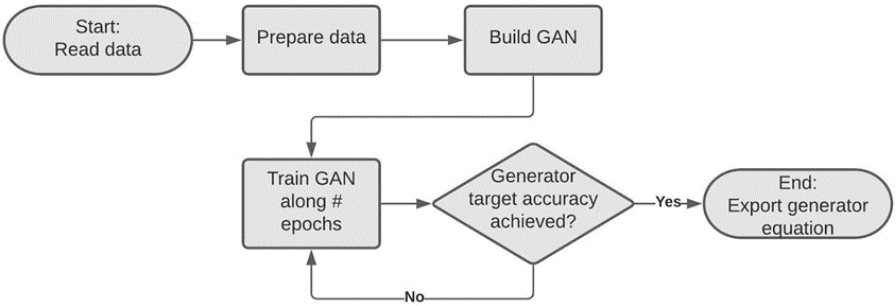


Figure 2: Proposed framework for using GANs to input data modeling.

The next step comprises the following data activities:

1. Remove observations with missing values.
2. Remove non-numeric columns.
3. Rescale input data.

This study assumes that the data intrinsic quality (Bokrants et al. 2015) is sufficient and, for this reason, no activities such as cleaning and outlier detection are necessary. Simulation researchers and practitioners should consider adopting other preparation activities if this assumption does not hold.

Data rescaling is an important step of data preparation and usually leads to more effective ML (Feng 2021). Generally, ML methods are more effective if data attributes have the same scale. This process will be helpful to speed up GAN training and to enhance the input model quality evaluation algorithm, a k-nearest neighbors (k-NN) classifier.

Several rescaling techniques can be adopted, such as min-max normalization and robust scaling (Raju et al. 2020). Min-max normalization rescales the dataset to a new limited range (e.g. [0, 1]). However, it will also limit the new generated data to the original data range, which may lead to input models with poor tail representation. In addition, min-max normalization is very sensitive to the presence of outliers. For these reasons, this study adopted the robust scaler. This technique removes the median and scales the data according to the interquartile range.

After data preparation, it is necessary to build the GAN components. Generator and discriminator were coded in Python using TensorFlow and Keras libraries. Our approach considers a "good" set of hyperparameters reported in the literature (Kurach et al. 2018). Among the defined GAN parameters, the following stand out, as presented in Table 1.

Once the GAN is set up, the training process starts. Batches of synthetic data (i.e., data generated by the generator) and real data are available to the discriminator, which focuses on learning to separate them (i.e., minimize its classification loss). Then, the generator is also trained, but with a different purpose: to generate batches of synthetic data that confuses the discriminator (i.e., maximize the discriminator classification loss). This iterative adversarial learning process is repeated until there are no remaining batches of real data, completing a learning epoch. Generally, GAN training requires several epochs until the generator reaches a suitable performance. For the present study, the number of epochs was set at 1,000.

Table 1: Main GAN parameters.

| Parameter | Description | Adopted values |
|---|---|---|
| Latent space size | Number of noise dimensions | 3 * number of dataset variables |
| Number of hidden layers | Number of layers between input and output layers | 2 |
| Hidden layers' density | Number of neurons per layer | 8 * number of dataset variables |
| Hidden layers' activation function | The function in an artificial neuron that delivers an output based on its inputs | Leaky rectified linear unit (He et al. 2015) |
| Output layers' activation function | | Linear for generator, and sigmoid for discriminator |
| Optimization algorithm for training and its learning rate | Algorithm used to iteratively optimize GAN weights to reduce losses | Adam (Kingma and Ba 2015). Learning rates: 1e-5 for the generator and 1e-4 for the discriminator |

The final generator performance should be measured to evaluate the input model accuracy. For this purpose, we adopted k-NN classifiers (Cover and Hart 1967), as proposed by Lopez-Paz and Oquab (2017). The classifier receives a dataset containing synthetic data and (all) real observations, in the same proportions, and tries to separate them. If synthetic data are perfectly realistic, the best decision the k-NN algorithm can make is to classify each observation at random. In this case, the classifier accuracy remains around chance level (50%). On the other hand, if synthetic data are not realistic at all, the classifier can easily separate real and synthetic observations, reaching accuracies of near 100%. In other words, a classifier accuracy of 50% can be interpreted as a 100% input model accuracy, while a 100% classifier accuracy can be interpreted as a 0% input model accuracy.

Finally, it is necessary to translate the input model to the language used by the adopted simulation software. Our algorithm has the capability to save generator network equations in a format compatible with FlexSim®.

## 4 RESULTS AND DISCUSSIONS

We evaluated the proposed method performance for four different distributions. Three of them comprise theoretical distributions, while one distribution is a real sample of cycle times in a Brazilian emergency department, as shown in Table 2. In the real case, patients arrive in the hospital; they are registered and go to triage. In the triage, they are classified according to their risk. After the registration, they are attended by a doctor and can be discharged or follow to further diagnosis and treatment. For this case, we considered triage and examination cycle times.

For each distribution, we carried out 10 replicates of the input modeling process. Each replicate was based on a different sample of the original distribution (sample sizes = 5,000). As an input model quality measure, we evaluated the input model accuracy (Lopez-Paz and Oquab 2017). In addition, based on the same samples, we built standard input models using ExpertFit®, a distribution fitting software, and compared the results with the GAN input model accuracy. As ExpertFit® only supports univariate distributions, variables in bivariate datasets were modeled independently.

Table 2: Distributions adopted for the method performance evaluation

| Distribution | Number of variables | Parameters | Total sample size | Sample size for input modeling |
|---|---|---|---|---|
| Normal | 1 | Mean = 100<br>Std. dev. = 3 | - | 5,000 |
| Exponential | 1 | Scale = 1 | - | |
| Bivariate normal | 2 | Mean = [100, 100]<br>Cov = [[3, 2.4],<br>[2.4, 3]] | - | |
| Real case: triage and examination cycle times | 2 | - | 101,531 | |

In the data preparation stage, observations of the real case dataset with missing data were removed. Moreover, all datasets were scaled using Robust Scaler. Then, for each replicate, the algorithm built GAN components (generator and discriminator) following parameters presented in Table 1. Figure 3 shows an example of these components' structures (layers and their input and output shapes).

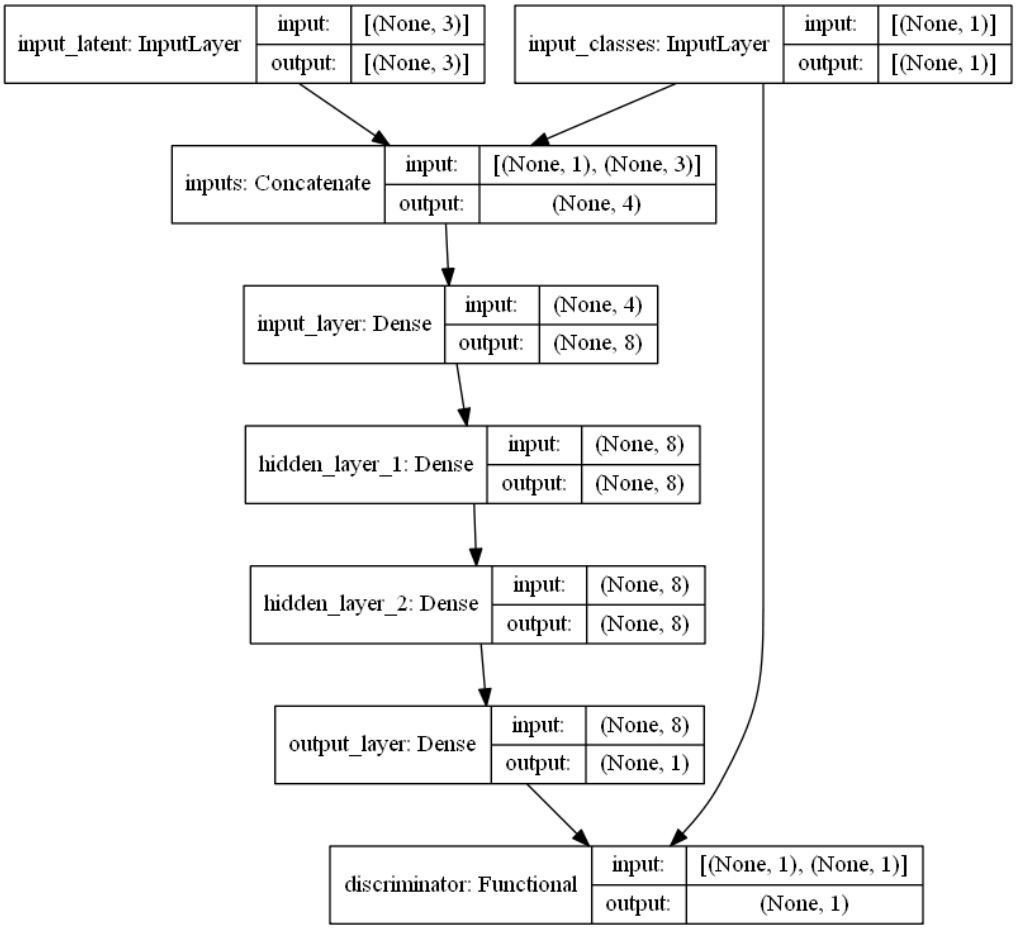


Figure 3: Generator network

For this study, generators include:

- 2 input layers: one for the latent variables and one optional layer for the desired data class definition (e.g., computer models may require conditional distributions, according to an entity characteristic). For more information on conditional GANs (cGANs), please refer to Mirza and Osindero (2014).
- 1 layer to concatenate both input layers.
- 3 dense layers: these layers are regular deeply connected neural network layers. They apply the activation function on the input and return the output.
- 1 output (dense) layer: this layer outputs synthetic samples.

On the other hand, discriminators comprise:

- 2 input layers: the discriminator receives as input both the real or synthetic sample and the observation classes.
- 1 layer to concatenate both input layers.
- 3 dense layers: similar to the generator.
- 1 output (dense layer): this layer outputs binary classification scores between 0 and 1. The larger the output, the greater the probability the evaluated observation is real.

In all cases, the number of epochs used to train GANs were set at 1,000. Generator performance (i.e., input model accuracy measured by k-NN classifier) was evaluated and tracked every 50 epochs. These checkpoints also saved model current parameters. As GAN results tend to oscillate during training, we selected the best tracked epoch and its model weights to use as input model rather than using the parameters of the last model checkpoint. Using histograms and scatterplots, Figures 4 and 5 show real (green) and synthetic (orange) distribution comparisons for the bivariate normal and real case distributions.

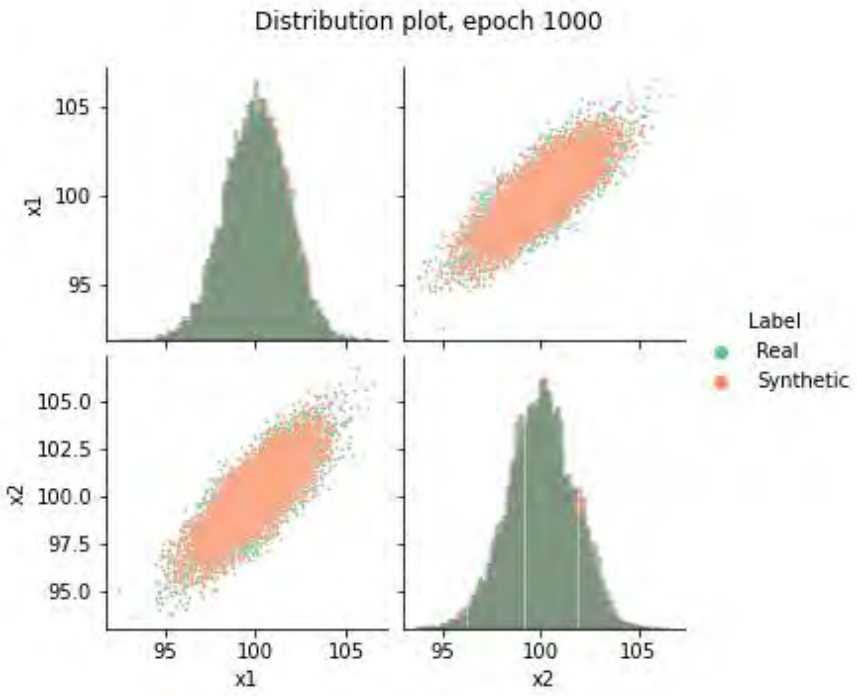


Figure 4: Real and synthetic bivariate normal distributions

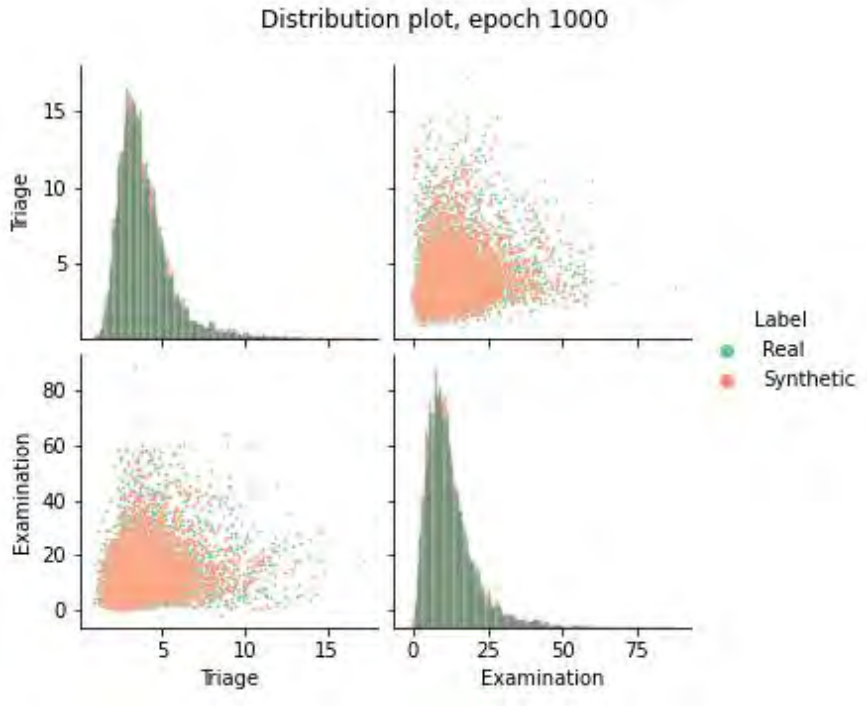


Figure 5: Real and synthetic real case distributions

These graphs visually indicate that input models developed using GANs can learn distributions characteristics such as shape and correlation.

Finally, the algorithm saved generator equations translated to the programming language used by FlexSim®, including latent space sampling, activation functions, and scaling.

We repeated the process for 10 replicates, to compare standard fitting and GAN fitting performances. For standard fitting of theoretical distributions, we estimated the probability distribution parameters according to each replicate input sample (e.g., mean and standard deviation for the normal distribution). For the real case, using ExpertFit®, we have chosen the best ranked probability distribution and its parameters. In all cases, the k-NN classifier was adopted to separate synthetic data and original data, returning input model accuracies. Figure 6 and Table 3 present the observed results.

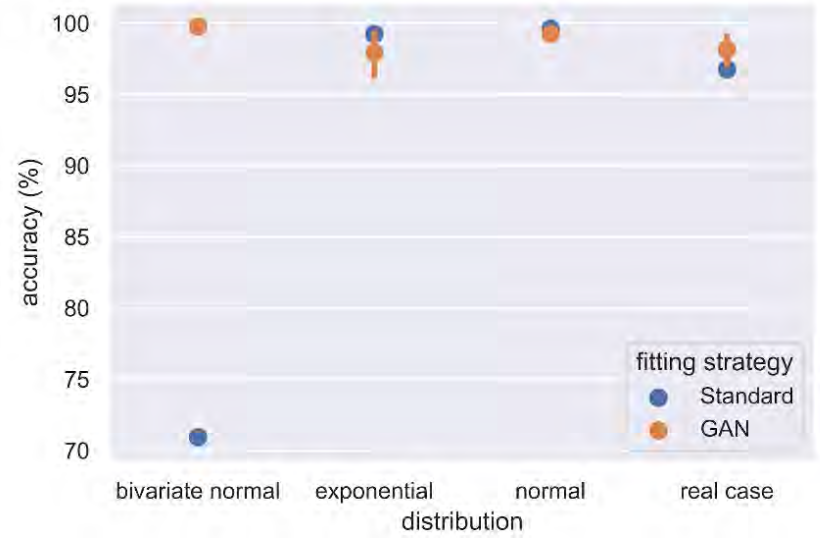


Figure 6: 95% confidence intervals for standard fitting and GAN fitting accuracies

Table 3: Distributions adopted for the method performance evaluation

| | Input model accuracy (%)[1] | |
|---|---|---|
| **Distribution** | **Standard fitting** | **GAN fitting** |
| Normal | 99.6 (99.4 - 99.9) | 99.3 (98.8 - 99.8)[2] |
| Exponential | 99.3 (99.0 - 99.6) | 97.9 (96.2 - 99.7)[2] |
| Bivariate normal | 70.9 (70.6 - 71.3) | 99.8 (99.5 - 100.0)[3] |
| Real case | 96.8 (96.4 - 97.2) | 98.2 (97.0 - 99.4)[3] |

1. Average accuracy and 95% confidence interval for 10 replicates. 2. No significant difference (2-sample t-test, p-value > 0.05). 3. GAN fitting accuracy significantly greater than standard fitting (2-sample t-test, p-value < 0.05)

GAN fitting achieved high average input model accuracies for all evaluated cases (average accuracies > 97.0%). For the normal and exponential distributions, there was no evidence of significant difference between standard and GAN fitting accuracy means. Nevertheless, it can be noted that GAN fitting accuracies presented larger variances. This may indicate that more than 1,000 epochs would be necessary for training. Further studies may evaluate and propose training stop criteria.

For the bivariate normal and real case distributions, there was evidence that GAN fitting may outperform standard fitting, especially considering the first one (+28.9 pp). These two cases have a special feature: data dependency. While the real case dataset presented a 0.153 Pearson's correlation coefficient, the bivariate normal distribution correlation was set at 0.800. This corroborates that GAN fitting may be a suitable option for modeling multivariate input data.

## 5 CONCLUSIONS

One of the main challenges to use simulation is the high time spent in its projects. Among all necessary steps to develop a simulation project, studies indicate that input data modeling is one of the most time-consuming. Although in this phase the main critical activities are related to data collection, we observe that this activity has been (and will be even more) facilitated with the advancement of technologies and solutions related to the new industry context. However, input data analysis will continue to consume

simulation practitioners' time and effort, especially considering digital twins that will require periodic revision. In this sense, we address a solution for complex data modeling, e.g., data with dependencies or incompatible with probability distributions offered by input modeling software. As already proven by previous studies, not considering data dependencies, or choosing an inaccurate input model may lead to unreliable conclusions.

Our study highlights how GANs may support input data modeling in simulation projects, especially in cases where traditional approaches are insufficient or inefficient. GAN input models were able to generate highly accurate synthetic samples (average accuracies > 97.0%). For univariate theoretical distributions, we found no significant difference between standard and GAN input models performances. On the other hand, for the real case distribution and the bivariate normal distribution, GAN input models outperformed standard ones. The most relevant accuracy gain was observed for the bivariate normal (+28.9 pp). It demonstrates that one of the main advantages of input modeling through GANs is the possibility of modeling joint distributions.

Future research may test other real cases and theoretical distributions, including different data types, such as integer and categorical. Moreover, further studies may compare GAN results with other fitting strategies (e.g., copula fitting and variational autoencoders). Finally, we suggest the evaluation of training stop criteria, different GAN architectures, including cGAN, and their parameter optimization.

## ACKNOWLEDGMENTS

The authors would like to express their gratitude to CNPq, CAPES, and FAPEMIG for their support throughout this research.

## REFERENCES

Agnese, J., J. Herrera, H. Tao, and X. Zhu. 2020. "A Survey and Taxonomy of Adversarial Neural Networks for Text-to-image Synthesis". *WIREs Data Mining Knowledge Discovery* 1: 1–26.

Akhavian, R., and A. H. Behzadan. 2014. "Knowledge-Based Simulation Modeling of Construction Fleet Operations Using Multimodal-Process Data Mining". *Journal of Construction Engineering and Management* 140 (2): 08013001.

Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol. 2010. *Discrete Event System Simulation*. 5. ed. New Jersey: Pearson.

Barlas, P., and C. Heavey. 2016. "Automation of input data to discrete event simulation for manufacturing: A review". *International Journal of Modeling, Simulation, and Scientific Computing* 7(1): 1–27.

Biller, B., AND C. Gunes. 2010. "Introduction to Simulation Input Modeling". In *proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 49-58. Baltimore, Maryland.

Bokrants, J., A. Skoogh, J. Andersson, J. Ruda, and D. Lämkull. 2015. "A Methodology for Continuous Quality Assurance of Production Data". 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, 1-13. Huntington Beach, California.

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

Cai, L., Y. Chen, N. Cai, W. Cheng, and H. Wang. 2020. "Utilizing Amari-Alpha Divergence to Stabilize the Training of Generative Adversarial Networks". *Entropy* 22: 1-19.

Cen, W., E. A. Herbert, and P. J. Haas. 2019. "Nim: Generative Neural Networks for Simulation Input Modeling". In *proceedings of the 2019 Winter Simulation Conference*, edited by N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, 1-2. National Harbor, Maryland.

Cover, T., and P. Hart. 1967. "Nearest Neighbor Pattern Classification". *IEEE Transactions on Information Theory* 13(1): 21-27.

Du, C., C. Du, and H. He. 2021. "Multimodal Deep Generative Adversarial Models for Scalable Doubly Semi-supervised Learning". *Information Fusion* 68: 118-130.

Feng, R. 2021. "Improving Uncertainty Analysis in Well Log Classification by Machine Learning with a Scaling Algorithm". *Journal of Petroleum Science and Engineering* 196: 107995.

Ferreira, W. P., F. Armellini, and L. A. Santa-Eulalia. 2020. "Simulation in Industry 4.0: A Atate-of-the-art Review". *Computers & Industrial Engineering* 149: 1-17.

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

He, K., X. Zhang, S. Ren, and J. Sun. 2015. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". *arXiv*, 1: 1-11.

Kingma, D., and J. Ba. 2015. "ADAM: A Method for Stochastic Optimization". In *proceedings of the 3rd International Conference on Learning Representations*, 1-15. San Diego, California.

Kuhl, M. E., E. K. Lada, M. A. Wagner, J. S. Ivy, N. M. Steiger, and J. R. Wilson. 2007. "Introduction to Modeling and Generating Probabilistic Input Processes for Simulation". In *proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 63-76. Austin, Texas.

Kurach, K., M. Lucic, X. Zhai, M. Michalski, and S. Gelly. 2018. "The GAN Landscape : Losses, Architectures, Regularization, and Normalization". *Proceedings of the Reproducibility in Machine Learning Workshop,* 1-16.Stockholm, Sweden.
Law, A. M. 2014. *Simulation Modeling and Analysis*. 5. ed. Boston: McGraw-Hill Science.
Leemis, L. 1997. "Seven Habits of Highly Successful Input Modelers". In *proceedings of the 1997 Winter Simulation Conference*, edited by S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, 39-46. Atlanta.
Lopez-Paz, D., and M. Oquab. 2017. "Revisiting Classifier Two-Sample Tests". In *proceedings of the 5th International Conference on Learning Representations*, 1-15. Toulon, France.
Mirza, M., and S. Osindero. 2014. "Conditional Generative Adversarial Nets". *arXiv*, 1: 1–7.
Montevechi, J. A. B., C. H. Santos, G. T. Gabriel, M. L. M. Oliveira, J. A. Queiroz, and F. Leal. 2020. "A Method Proposal for Conducting Simulation Projects in Industry 4.0: A Cyber-Physical System in an Aeronautical Industry". In *proceedings of the 2020 Winter Simulation Conference*, edited by K. -H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 2731-2742. Online.
Mourtzis, D. 2020. "Simulation in the Design and Operation of Manufacturing Systems: State of the Art and New Trends." *International Journal of Production Research* 58 (7): 1927–1949.
Pan, Z., W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng. 2019. "Recent Progress on Generative Adversarial Networks (GANs): A Survey". *IEEE Access* 7: 36322-36333.
Raju, N., K. Lakshmi, V. Scholar, A. Kalidindi, and V. Padma. 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.
Robertson, N., and T. Perera. "Automated data collection for simulation?". *Simulation Practice and Theory* 9:349–364.
Robinson, S. 2004. *Simulation: the practice of model development and use*. 1. ed. Chichester: Wiley.
Rodič, B. 2017. "Industry 4.0 and the New Simulation Modelling Paradigm". *Organizacija* 50(3): 193-207.
Sargent, R. G. 2013. "Verification and validation of simulation models". *Journal of Simulation* 7(1): 12–24.
Saez, M., F. P. Maturana, K. Barton, and D. M. Tilbury. 2018. "Real-Time Manufacturing Machine and System Performance Monitoring Using Internet of Things." *IEEE Transactions on Automation Science and Engineering* 15 (4): 1735–1748.
Scheidegger, A. P. G., T. F. Pereira, M. L. M. Oliveira, A. Banerjee, and 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.
Scholl, W., D. Noack, O. Rose, B. P. Gan, M. L. Peh, P. Lendermann, and P. Preuss. 2010. "Towards Realization of a High-Fidelity Simulation Model for Short-Term Horizon Forecasting in Wafer Fabrication Facilities." *In proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan. 2563–2574. Baltimore, Maryland.
Skoogh, A., and B. Johansson. 2008. "A Methodology for Input Data Management in Discrete Event Simulation Projects". In *proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler, 1727-1735. Miami, Florida.
Skoogh, A., and B. Johansson. 2009. "Mapping of Time-Consumption during Input Data Management Activities". *Simulation Notes Europe* 19(2): 39–46.
Skoogh, A., T. Perera, and B. Johansson. 2012. "Input data management in simulation – Industrial Practices and Future Trends". *Simulation Modelling Practice and Theory* 29: 181–192.
Sorin, V., Y. Barash, E. Konen, and E. Klang. 2020. "Creating Artificial Images for Radiology Applications Using Generative Adversarial Networks (GANs) – A Systematic Review". *Academic Radiology* 27(8): 1175-1185.
Sormaz, D. N., and M. Malik. 2018. "Data-DrivenSimulation Modelling for Progressive Care Units in Hospitals". *Procedia Manufacturing* 17: 819–826.
Tolosana, R., R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia. 2020. "Deepfakes and Beyond : A Survey of Face Manipulation and Fake Detection". *Information Fusion* 64: 131–148.

## AUTHOR BIOGRAPHIES

**JOSÉ ARNALDO BARRA MONTEVECHI** is a Titular Professor of 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 e-mail address is montevechi@unifei.edu.br.

**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 e-mail address is afonso.teberga@gmail.com.

**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 Healthcare Systems. His e-mail address is gustavo.teodoro.gabriel@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 e-mail address is chenrique.santos@unifei.edu.br.