Generative adversarial networks to model air pollution under uncertainty

Jamal Toutouh[1], **Sergio Nesmachnow**[2], **and Diego Gabriel Rossit**[3]

[1] Massachusetts Institute of Technology, USA
[2] Universidad de la Repblica, Uruguay
[3] INMABB, DI, Universidad Nacional del Sur (UNS)-CONICET, Argentina

E-mail: toutouh@mit.edu, sergion@fing.edu.uy, diego.rossit@uns.edu.ar

Abstract. Urbanization trends worldwide show a clear preference for motorized road mobility, which has led to a degradation of air quality in recent years. Modelling and forecasting ambient air pollution is a relevant problem because it helps decision-makers and urban city planners understand this phenomenon, which is a significant threat to citizens' health. Generally, data-driven models suffer from a lack of data. This article addresses the issue of having limited access to road traffic density and pollution concentration data by applying deep generative models, specifically, Conditional Generative Adversarial Networks (CGAN). The main idea is to train CGANs to generate synthetic nitrogen dioxide concentration values given the road traffic density. The experimental data analysis from Montevideo (Uruguay) shows that the proposed method generates realistic (accurate and diverse) pollution data while using reduced computational resources.

1. Introduction

The growth of the cities that prioritized motorized mobility (use of the individual or collective vehicles) is having an undesired negative effect on the dwellers' safety and quality of life. A significant concern is the high generation of emissions (air pollutants) due to the rapid development of car-oriented cities. Air pollution is a major concern because it has a negative impact on the citizens' health, e.g., it provokes several respiratory diseases and it reduces life expectancy [1, 2].

One of the most important sources of air pollutants in urban areas is road mobility [1, 2]. Thus, proposing mobility policies that reduce road traffic (i.e., the use of private vehicles) could be an effective strategy to mitigate the generation of emissions and improve urban livability and inhabitants' health. However, it is not easy to understand the various phenomena that may have implications for the production or dissipation of pollutants, e.g., weather or time of the day. Even, the policy-makers may see these kinds of measure as a way to degrade the effectiveness and efficiency of road transportation. For this reason, there have been different approaches to evaluate the real impact of mobility policies on the air quality [3]. Thus, modelling, predicting, and forecasting ambient air pollution allow policy-makers and urban city planners to provide solutions to this issue.

Artificial Neural Networks (ANN) and Deep Learning (DL) are successfully applied to deal with air outdoor pollution modelling, prediction, and forecasting, as data-driven methods [4]. On the one hand, the main advantage of this approach is that the use of ANNs does not require an in-depth understanding of the physics and dynamics between air pollution concentration levels and other explanatory variables. On the other hand, these kinds of approaches have a set of open questions: the selection of the appropriate ANN model, the interpretation of the results of that kind of black-box methods, and the results are problem-specific. Besides, it is a significant matter that this kind of methods requires a vast amount of data to be trained [4].

This article focuses on training generative models, as a data augmentation approach, to produce new information units (levels of NO_2) to feed data-driven ANN methods for modelling, forecasting, and predicting outdoor pollution. Generative Adversarial Networks (GANs) are successfully used to train generative models [5] to learn to represent an estimate of a data distribution given by the training dataset. Thus, we propose the use of a specific type of GANs, Conditional GANs (CGANs), to train generators able to create synthesized pollution data given the road traffic volume. The real dataset used to train the CGANs is built by collecting the levels of NO_2 and road traffic volume gathered by sensors located in Montevideo (Uruguay). Thus, the CGAN will produce new information units (levels of NO_2) that approximate the original training set. Notice that we are not trying to create a pollution forecasting method, but a modelling one from training the generative models. A previous study applied a similar approach to model pollution [6]. However, the authors did not consider the road traffic, which is one of the primary sources of NO_2 in urban environments.

2. Conditional GANs for pollution modeling

GANs consist of two artificial neural networks, a generator and a discriminator, that apply adversarial learning. The generator is trained to deceive the discriminator by generating "fake" or "artificial" data samples transforming its inputs from a random latent space. The discriminator learns how to distinguish between the "real" and "artificial" data samples. GAN training is formulated as a minimax optimization problem by the definitions of generator and discriminator loss [5]. CGANs are an extension of GANs to deal with labeled training datasets (structured in classes, i.e, each sample has a y label). The idea is to train generative models able to create samples of a given class according to y.

The general training of a CGAN is graphically described in Figure 1.

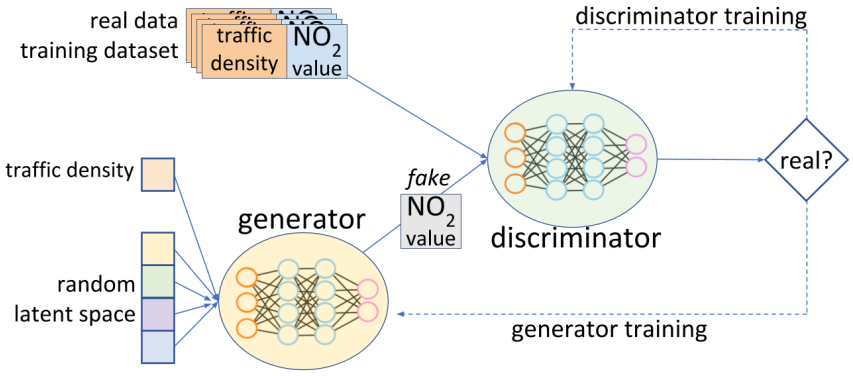


Figure 1. General CGANs training setups.

The proposed approach applies a CGAN to train a generative model to get spatial pollution data. The CGAN learns the probability distributions of the generated pollutant given the road traffic density (given by a class from 1 to 5, from lowest to highest). Thus, the generative model has the road traffic density class as an input and returns the predicted NO_2 concentration, which is a random value drawn from the probability distribution learned by the generator.

As the CGAN training is defined as minimax optimization between the generator and the discriminator, the training process may oscillate without converging to an equilibrium. Thus, our method tracks the accuracy of the generator after each training epoch in terms of distance between the real and the generated distributions, and keeps a copy of the best generator found.

3. Experimental setup

This section summarizes the main details of the methodology for training CGANs to create synthetic pollution data.

3.1. Training dataset

The training dataset studied here is provided by the open data portal offered by the National Government of Uruguay [7]. Specifically, the training dataset consists of NO_2 concentration in micrograms per cubic meter ($\mu g/m^3$) and road traffic density in vehicles per hour gathered in three different locations in Montevideo (i.e., *Avda. Colon*, *Calle Tres Cruces*, and *Curva de Maronas*) during January–February 2020 (in which there are several periods without data). The data is hourly averaged.

The road traffic density is classified in five classes: A, B, C, D, and E, which represent densities from very low to very high. Table 1 and Figure 2 summarizes the distributions of data used as training dataset. As can be seen, the NO_2 concentration increases with the road traffic volume. The classes are highly unbalanced (see Table 1), i.e., A class has 1280 (the maximum) and C class has 539 samples (the minimum). We randomly sampled over the classes for our experiments to select 539 samples of each class to balance the dataset to avoid training biases. Thus, the final training dataset size is 2695 (539×5).

traffic density class	mediean	iqr	number of samples
A - very low traffic	18.50	23.00	1280
B - low traffic	28.00	33.00	743
C - fluid flow traffic	32.00	41.75	539
D - high traffic	40.50	48.00	626
E - very high traffic	49.00	50.00	650

Table 1. Summary of the NO_2 pollution distributions (in terms of median and interquartile range) and the number of samples for each traffic density class.

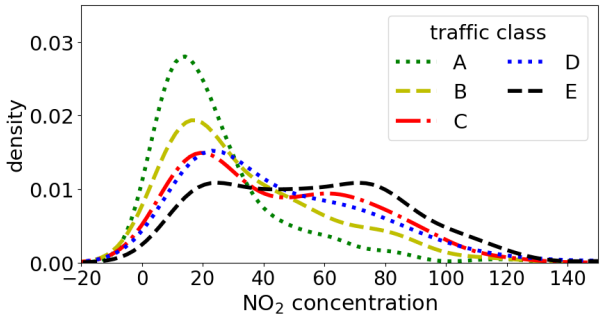


Figure 2. Training dataset, i.e., pollution data distribution given road traffic class.

3.2. CGAN design details

In our research, both ANNs, the generator and the discriminator, are implemented as multilayer perceptrons (MLP) [8]. Our experiments explore the use of three different MLP architectures for the generator and the discriminator in the proposed CGAN: Three different neural networks architectures for the generator and the discriminator in the proposed CGAN: CGAN-1, three-layer perceptron (256 neurons per hidden-layer); CGAN-2, four-layer perceptron (128 neurons

per hidden-layer); and CGAN-3, four-layer perceptron (256 neurons per hidden-layer). The main idea is to analyze the impact on the efficacy and the efficiency of the CGAN training is affected by the complexity of the ANNs (number of parameters).

For all the CGANs evaluated, the generators' input layer has size 65 (64 for reading the random latent space and one for the road traffic density label). The output layer has size one (for the predicted NO_2 concentration). The input layer of the discriminator has size two (one for the road traffic density label and one for the NO_2 concentration), and the output layer has size one (for the predicted label, i.e., *fake* or *real*).

3.3. Metrics evaluated

The main goal of the experimental evaluation is to analyze the impact on the training, the quality of the results, and the computational cost of the different CGAN architectures.

The performance is evaluated according to the loss values computed for the generator and the discriminator during the training process. In this research, the function applied to compute the loss is the binary cross-entropy (BCE) [5]. Equations 1 and 2 present the discriminator and generator loss, respectively.

$$\mathcal{L}_d = \frac{1}{2}\mathbb{E}_{x,y\sim P_{data}(x,y)}[log(D_d(x,y))] - \frac{1}{2}\mathbb{E}_{z\sim P_z(z),y\sim P_y(y)}[log(1 - D_d(G_g(z,y),y))] \,, \quad (1)$$

$$\mathcal{L}_g = \mathbb{E}_{z\sim P_z(z),y\sim P_y(y)}[log(1 - D_d(G_g(z,y),y))] \quad (2)$$

In order to assess the quality of the generated samples, we evaluate the distance between the *real* and *syntezised* data distributions (i.e., NO_2 concentration values) for each road traffic density class. We propose the use of the distance between distributions according to the KolmogorovSmirnov statistical test. Thus, lower values indicate better sample quality.

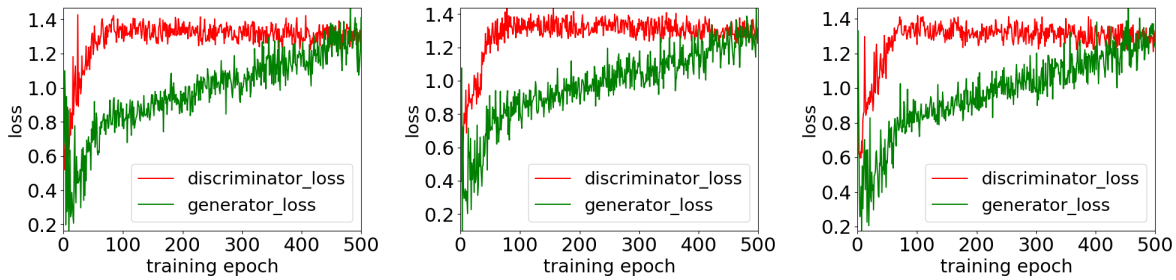
Finally, we also consider the computational time and the training epoch the generator created the most accurate pollution distribution (*iteration best found*) to evaluate the computational cost of the proposed generative methods.

4. Experimental analysis

In order to perform the numerical analysis of our approach, the three CGAN approaches have been configured with a learning rate of 0.0002, batch size of 25 samples (108 batches per training epoch), and 500 training epochs. This section presents the results of performing 30 independent runs for each CGAN training in National Supercomputing Center (Cluster-UY), Uruguay [9].

Figure 3 illustrates the evolution of the loss of the generator and the discriminator during the training process. The three approaches show similar behaviour. During the first 100 epochs, both losses oscillate and increase. This behaviour is explained by the fact that the discriminator is not trained enough to give the proper feedback required by the generator to learn. After that, the discriminator starts becoming stronger and reduces the loss values. Thus, it is harder for the generator to deceive the discriminator. For this reason, the generator begins increasing the computed loss values, allowing the generator to learn how to create more accurate samples. At the end of the training process, the generator and the discriminator have similar loss values, which could indicate that they have reached an equilibrium.

Table 2 summarizes the *fake* data distributions by showing the median and the interquartile range (iqr) and Figure 4 illustrates the distributions. The median values of the NO_2 generated distributions are very close to the *real* ones. However, the diversity (in terms of iqr) of the *fake* distributions is substantially lower than in *real* data. Figure 4 confirms that the *fake* distributions are less dispersed than the real ones in Figure 2. This indicates that our methods are able to capture the general behaviour of the NO_2 concentration given the traffic class, but it has some limitations to capture the whole *real* distribution. Different authors have proposed methods to improve diversity in GAN training [10, 11, 12].

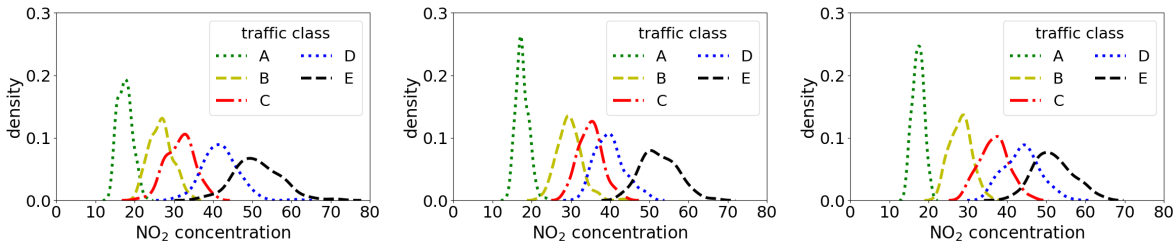


(a) CGAN-1 (b) CGAN-2 (c) CGAN-3

Figure 3. Discriminator and generator loss values through the training process.

traffic density class	*real data*		CGAN-1		CGAN-2		CGAN-3	
	mediean	iqr	mediean	iqr	mediean	iqr	mediean	iqr
A - very low traffic	18.50	23.00	17.30	2.03	17.41	2.16	**17.95**	2.69
B - low traffic	28.00	33.00	**28.33**	3.75	29.55	3.67	28.76	4.26
C - fluid flow traffic	32.00	41.75	36.68	5.12	34.83	4.25	**31.83**	4.95
D - high traffic	40.50	48.00	43.90	5.98	**39.76**	4.97	41.26	5.58
E - very high traffic	49.00	50.00	51.26	6.48	52.72	6.46	**49.70**	7.84

Table 2. Summary of the NO_2 pollution distributions synthesized by the computed generative models in terms of median and interquartile range.



(a) CGAN-1 (b) CGAN-2 (c) CGAN-3

Figure 4. Generated pollution distributions given road traffic class by the proposed CGANs.

Table 3 reports the minimum (min), mean, and standard deviation (std) of the distance between the synthesized data and the real data distributions (*min distance*), the computation time of each independent run (*computational time*), and the training epoch the generator created the most accurate pollution distribution (*iteration best found*).

CGAN type	*min distance*			*computational time*			*iteration best found*		
	min	mean	std	min	mean	std	min	mean	std
CGAN-1	**0.41**	**0.43**	0.01	**318.41**	**677.11**	179.07	21.00	234.61	140.40
CGAN-2	0.42	**0.43**	0.01	318.99	827.67	173.34	51.00	165.13	85.04
CGAN-3	**0.41**	**0.43**	0.01	341.83	859.05	106.43	**20.00**	**146.10**	96.90

Table 3. Experimental results in terms of best generator found and computational cost.

In terms of the distance between the generated distributions and the real data (see Table 3), the three proposed CGANs create accurate distributions with the same quality (non-significant differences between them). This is in line with the results showed in Table 2 and Figure 4.

Focusing on the computational cost, on the one hand, CGAN-1, the smallest model, shows the shortest run times. On the other hand, CGAN-3 finds the best generators faster than the others. These results are expected because when training ANNs with the same number of iterations, the computational time increases with the ANN's complexity (i.e., with the number of parameters to train). In turn, bigger networks are able to capture more complex features requiring a lower number of training epochs.

5. Conclusions and future work

We have proposed using CGANs to train generative models to create synthesized pollution data, in this case, NO_2 concentration, according to a given road data traffic density. The idea is to deal with the lack of data suffered by data-driven methods for modelling, predicting, and forecasting ambient air pollution.

We have proposed three different CGANs according to the complexity of the ANNs architectures used for the generator and the discriminator. The main results indicate that the three proposed models generate accurate NO_2 pollution while requiring a reduced computational time. All the CGANs have shown robustness on the training because all the experiments converged to accurate generators. However, these generative models are limited in terms of producing diverse data samples.

The main lines for future work are related to extend the proposed model to generate the pollution of the whole city of Montevideo by taking into account information from more sensors, defining the generative modelling problem taking into account other variables such as the weather or the time, and applying the generated data to feed data-driven models to prove that they are able to improve their accuracy after including fake samples.

Acknowledgements

This research was partially funded by European Unions Horizon 2020 research and innovation program under the Marie Skodowska-Curie grant agreement No 799078, by the European Union H2020-ICT-2019-3 and UMA18-FEDERJA-003, and the Systems that Learn Initiative at MIT CSAIL.

References

[1] Lebrusán I and Toutouh J 2020 *Smart Cities* **3** 456–478
[2] Lebrusán I and Toutouh J 2020 *Air Quality, Atmosphere & Health* **14**(3) 333–342
[3] Toutouh J, Lebrusán I and Nesmachnow S 2020 Computational Intelligence for Evaluating the Air Quality in the Center of Madrid, Spain *International Conference on Optimization and Learning* (Springer) pp 115–127
[4] Cabaneros S M, Calautit J K and Hughes B R 2019 *Environmental Modelling & Software* **119** 285–304
[5] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial nets *Advances in neural information processing systems* pp 2672–2680
[6] Toutouh J 2021 Conditional generative adversarial networks to model urban outdoor air pollution *Smart Cities* ed Nesmachnow S and Hernández Callejo L (Cham: Springer International Publishing) pp 90–105 ISBN 978-3-030-69136-3
[7] Gobierno Abierto de AGESIC 2020 Catlogo Nacional de Datos Abiertos `https://catalogodatos.gub.uy/` Accessed: 2020-10-30
[8] Hastie T, Tibshirani R and Friedman J 2009 *The Elements of Statistical Learning* (Springer New York)
[9] Nesmachnow S and Iturriaga S 2019 Cluster-UY: Collaborative scientific high performance computing in uruguay *Supercomputing* pp 188–202
[10] Toutouh J, Hemberg E and O'Reilly U M 2019 Spatial evolutionary generative adversarial networks *Proceedings of the Genetic and Evolutionary Computation Conference* GECCO '19 (New York, NY, USA: ACM) pp 472–480 ISBN 978-1-4503-6111-8
[11] Toutouh J, Hemberg E and O'Reily U M 2020 Re-purposing heterogeneous generative ensembles with evolutionary computation *Proceedings of the 2020 Genetic and Evolutionary Computation Conference* GECCO '20 (New York, NY, USA: Association for Computing Machinery) p 425434 ISBN 9781450371285
[12] Toutouh J, Hemberg E and O'Reilly U M 2020 *Data Dieting in GAN Training* (Singapore: Springer Singapore) pp 379–400 ISBN 978-981-15-3685-4