Generative neural networks for synthetic data generation in insurance: context and use cases

Paris, 17/11/2022

Hadrien COLVEZ[1], Aurélien COULOUMY[2], Akli KAIS[2]

With the kind contributions of: Mohammed Amine BEN CHEIKH LEHOCINE[2] and Eric LAVERGNE[2]

1. Seabird (hcolvez@seabirdconseil.com)
2. CCR Group (acouloumy@ccr.fr)

1. INTRODUCTION

Problems with data

1.1 PROBLEMS WITH DATA

- **Data are key drivers** for insurers but require sourcing, labelling, budget, etc.
- Plus, they are **not always what we would expect**, and it may cause troubles:

Quality

- Difficulty to produce accurate calculations
- Additional work for controls, imputation tools or data acquisition

Exhaustivity

- Model miss training
- Tool miss testing
- Inability to interpret adversarial cases or drift
- Require more labeling

Privacy

- Exposure to Personal or Medical Data legal constraints (GDPR)
- Contractual or strategical concerns

1.2 USE SYNTHETIC DATA

- **Synthetic data** aim at generating **"fake data"** that are similar to data from real world.
- It may be well suited for insurance use cases:

Figure 1: 3D aerial image generation Bifrost.ai

Quality	- Missing data, incoherent values, ineffective data quality techniques for law behaviour setup, technical pricing or reserving calculations.	Data **imputation**
Exhaustivity	- Limited labelling budget, lack of data regarding emerging risks, new stress test for capital modelling, rare events scenario for natcat, fraud.	Data **augmentation**
Privacy	- Restricted use of medical or geotracked data for actuarial calculations, HDS storage, third parties (broker, MGA) share.	Data **anonymisation**

2. WHAT ARE SYNTHETIC DATA?

Approaches and methods

2.1 HOW TO GENERATE SYNTHETIC DATA?

- Generated data samples must have the **same statistical** / **structural** properties as real data. Two main data synthesizer approach exist: sampling and simulation based methods.

- About **sampling-based** techniques:

  - Fit statistical estimators or train machine learning models on real data to learn an approximate distributions

  - Infere to get samples of new synthetic data

- Sampling-based methods can be used on **any type of data** (tabular, images, time series)

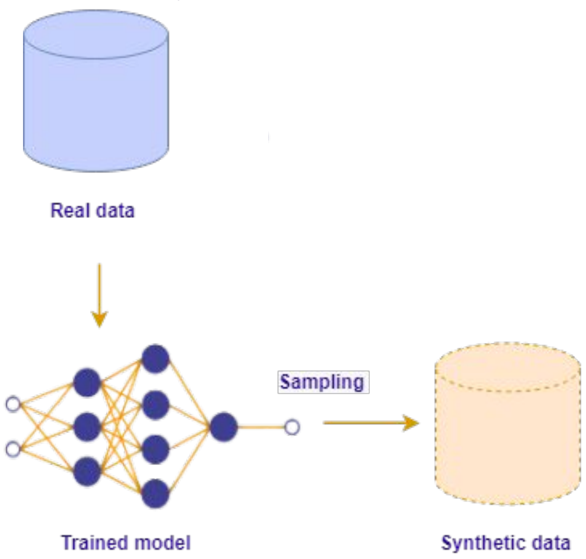


Figure 2: Sampling-based method scheme

2.2 SYNTHETIC DATA GENERATION FAMILIES

- Several sampling-based methods have been developed in recent years going from simple **statistical methods** to more complex techniques using **neural networks**:

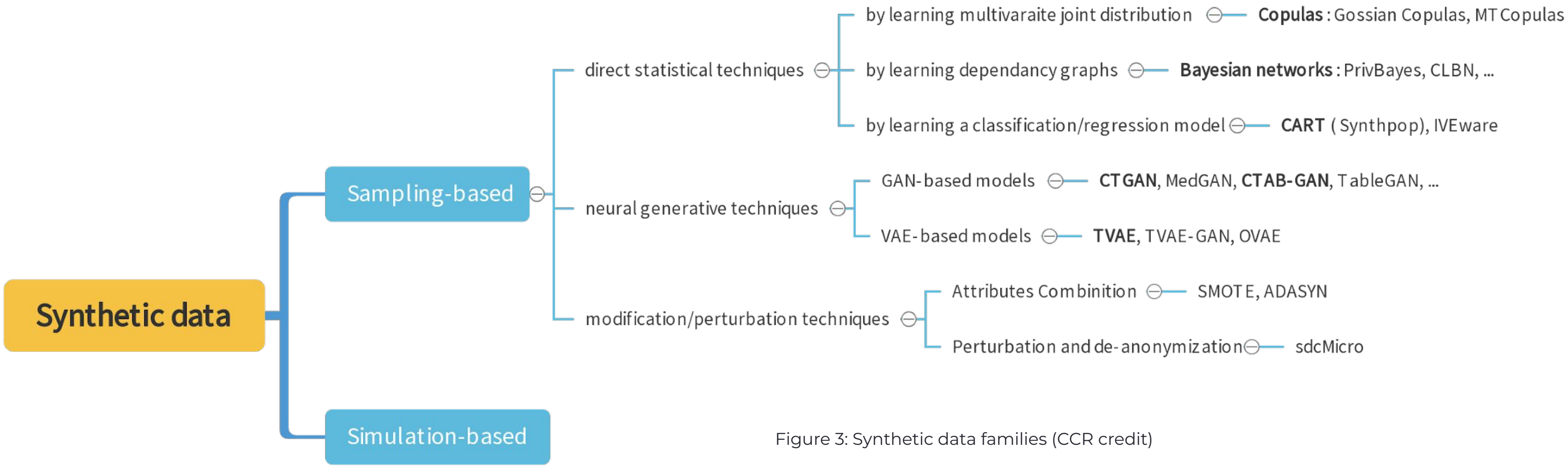


Figure 3: Synthetic data families (CCR credit)

2.3 NEURAL GENERATIVE TECHNIQUES (GAN)

- Generative Adversarial Networks (GANs) are **deep learning** models based on **adversarial training** that can learn to **generate** new samples of content.

  - The primary objective of the GANs is to learn the unknown **probability distribution** of the data

  - Composed of two architectural components: a **generator** and **discriminator**

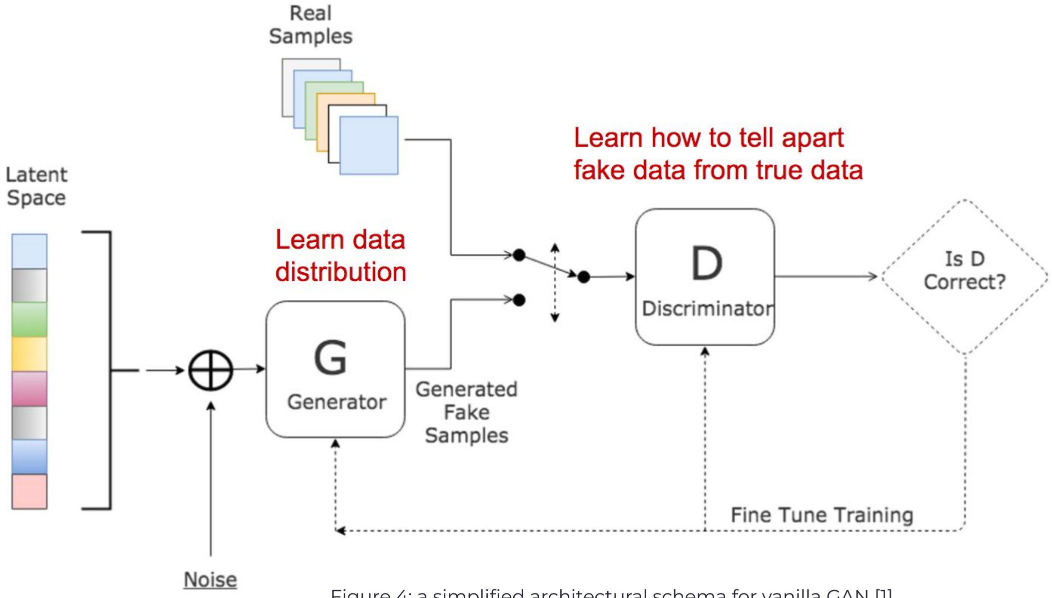


Figure 4: a simplified architectural schema for vanilla GAN [1].

[1] PEIXIANG Z., Generative Adversarial Network (GAN) Overview, 2018, Github, [Link]

2.3 NEURAL GENERATIVE TECHNIQUES (GAN)

- **Generation phase**: goal is to make **fake** data looks **similar** to the one we get from real events.

- **Discrimination phase:** it **classifies** fake data from the generator.

- **Final phase:**

  - When the discriminator's accuracy **reaches 50%**, it is no longer possible for the discriminator network to distinguish real from fake samples.

  - The generated samples **are similar to** those obtained from real world.

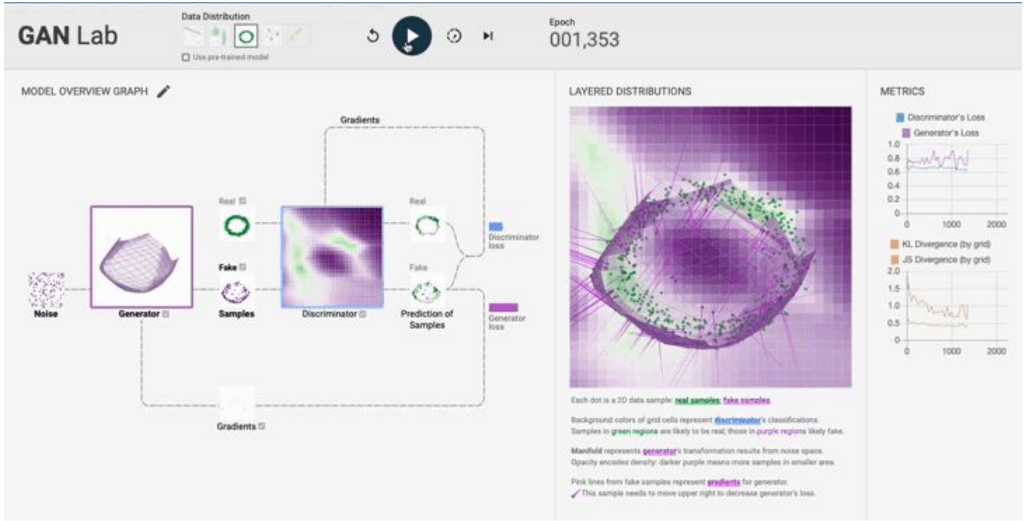


Figure 5: demo on how GANs works by GAN Lab [2]

[2] Minsuk Kahng, Nikhil Thorat, Polo Chau, Fernanda Viégas, and Martin Wattenberg. "GAN Lab: Understanding Complex Deep Generative Models using Interactive Visual Experimentation." Jan. 2019. https://arxiv.org/abs/1809.01587

2.4 CONDITIONAL GAN

- Conditional GAN on Tabular Data (CTGAN), is an **adaptation** of GAN architecture to model **tabular** data using a **conditional** generator

  - Extension of vanilla GAN by conditioning both the generator and the discriminator with an **extra** information

  - Augments the training procedure taking into account **data imbalance** through use of a conditional generator and sample training for discrete column

  - Uses a preprocessing step called **mode-specific normalization** to normalize continuous columns

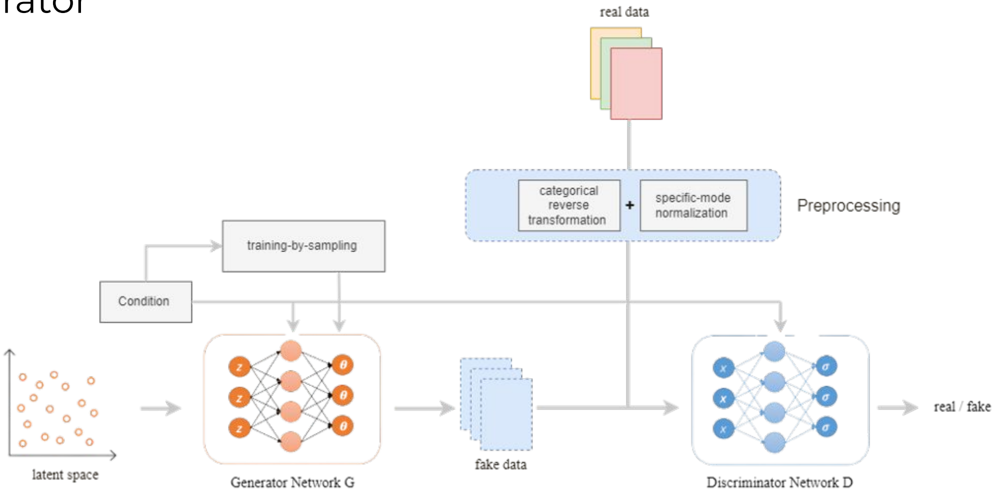


Figure 6: Architecture of a CTGAN model [3].

[3] Lei et al, Oct 2019. Modeling tabular data using conditional GAN. https://arxiv.org/abs/1907.00503

3. HOW TO USE SYNTHETIC DATA ?

Use cases and implementation

3.1 QUALITY - MOTOR PRICING

- Data provided by an insurance pricing game competition*.
- Nearly **60k** real historical **motor insurance** policies for 4 consecutive years.
- Each policy concerns a vehicle, its drivers and an accident history over 4 years with a total of **228k** observations.

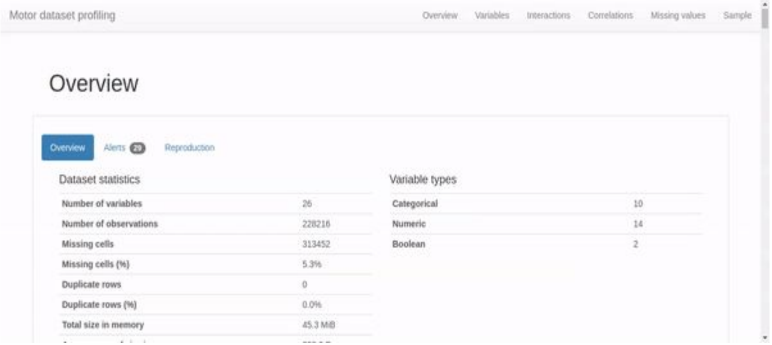


Figure 7: Data quality overview using pandas profiling

- **Key features**: vehicle age, vehicle value, speed, driver age, license age, coverage of policy, policy duration, etc.

Are synthetic data methods relevant for missing values imputation ?

*AIcrowd https://www.aicrowd.com/challenges/insurance-pricing-game

3.1 QUALITY - SYNTH. INPUTATION STRATEGIES

- **Method A - Univariate**
  (Synthetic Data + Simple desc. stat. imputer)
- **Method B - Multivariate**
  (Synthetic Data + Similarity matching)
- **Method C - Multivariate**
  (Synthetic Data + MissForest[4] imputer)

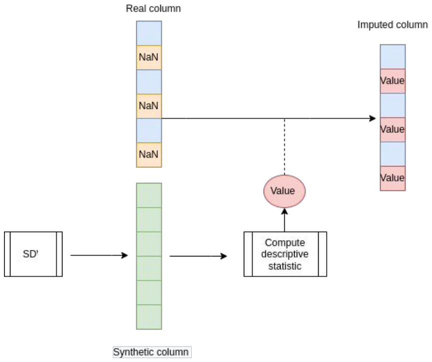


Figure 8: Method A

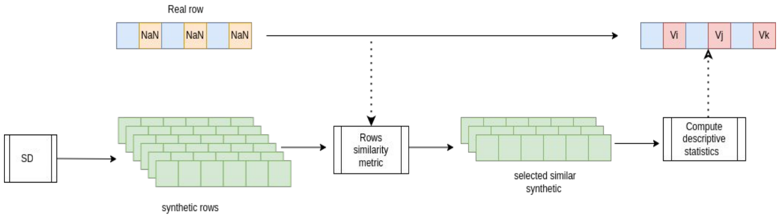


Figure 9: Method B

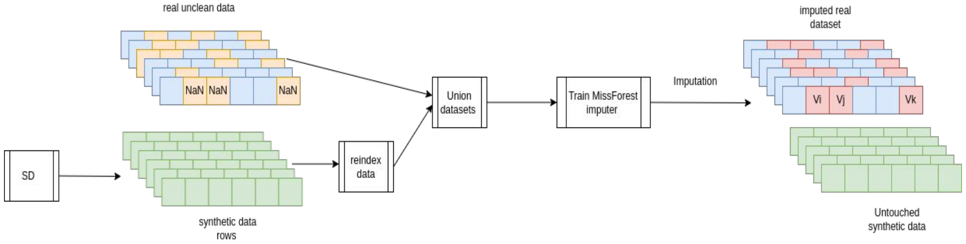


Figure 10: Method C

[4] Daniel J. Stekhoven, Peter Bühlmann, MissForest—non-parametric missing value imputation for mixed-type data, https://doi.org/10.1093/bioinformatics/btr597

3.1 QUALITY - EVAL. METRICS

- We evaluate methods by creating a **corrupted datasets**:

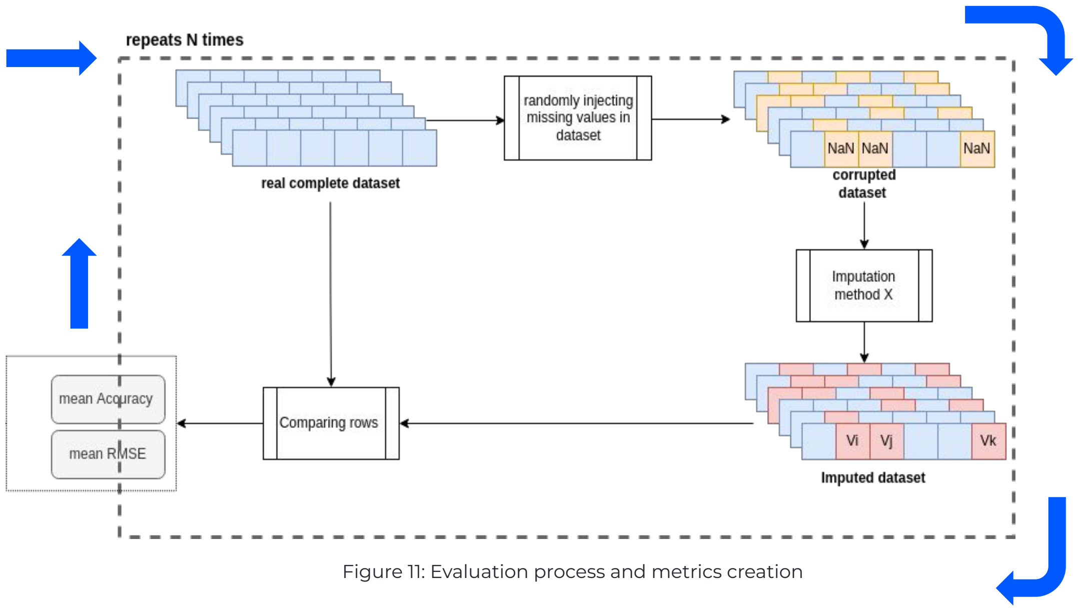
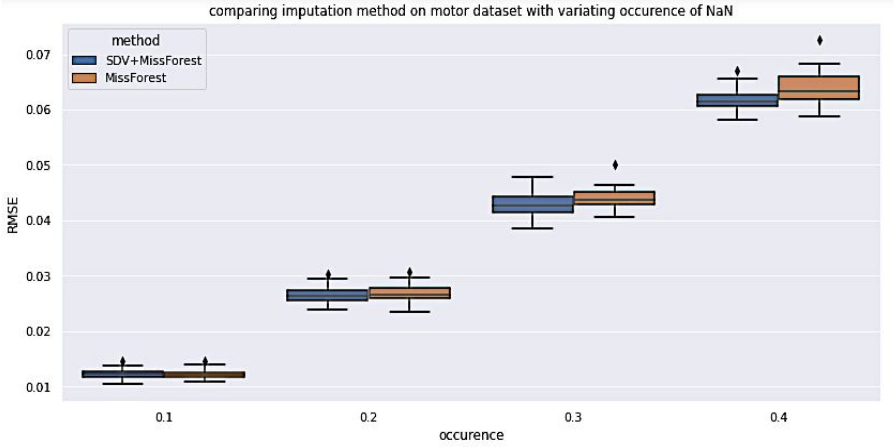


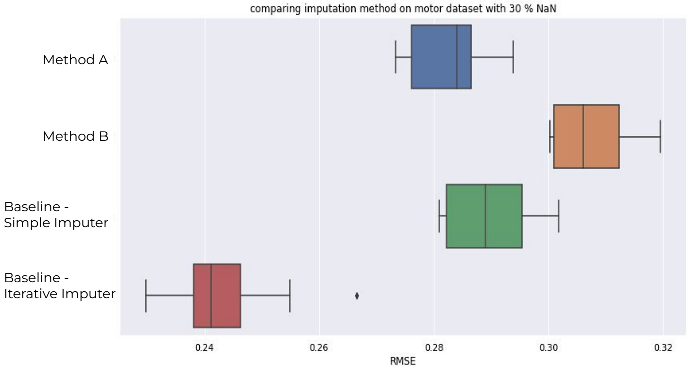
Figure 11: Evaluation process and metrics creation

- We **loop through** this process to get a metrics distribution per method

3.1 QUALITY - IN PRACTICE

- Strategies based on **MissForest** model **give higher performances**
- Using data augmentation with MissForest **becomes relevant** when frequency of NaN **is high (>20%)**





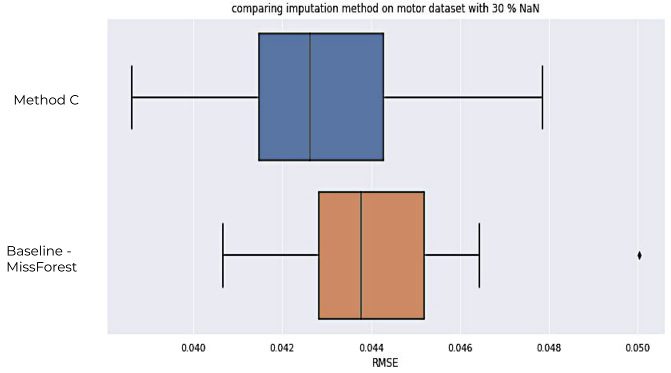
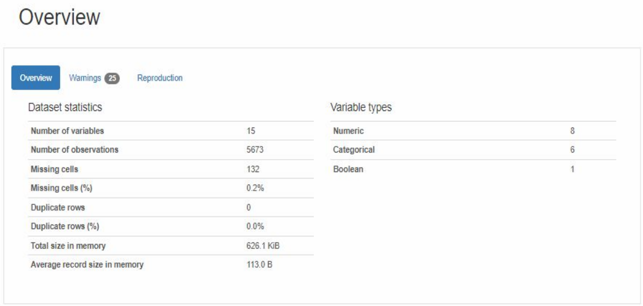


Figure 12: RMSE Box plot for different synthesizer techniques

3.2 EXHAUSTIVITY - CLAIMS ANALYSIS

- **French motor insurance** portfolio collected for reinsurance purpose.
- **~2k severe bodily injury** claims from 1999 to 2021, reviewed annually.
- Updated prejudices charges with **~137k observations.**
- **Key features identified:** age, sex and socio-professional category of the victim, type of injury, rate of permanent damage to physical integrity.



Overview

Overview | Warnings 25 | Reproduction

Dataset statistics

Number of variables	15
Number of observations	5673
Missing cells	132
Missing cells (%)	0.2%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	626.1 KiB
Average record size in memory	113.0 B

Variable types

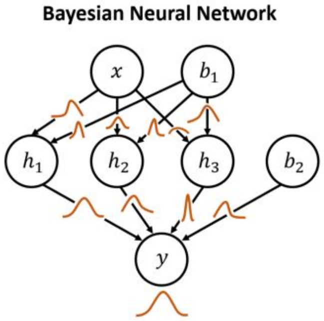
Numeric	8
Categorical	6
Boolean	1

Figure 13: Data quality overview using pandas profiling

? How to use **synthetic data** methods to **improve model knowledge?**

3.2 EXHAUSTIVITY - UNCERTAINTY AND ADVERSARIAL DATA

- **ML** models may not be trained and tested on the **whole** observations possibilities
- Techniques such as **BNN** [5] are helpful because they introduce uncertainty [6] measures that point knowledge **weakness** but not **unknown** scenarios
- Synthetic data are used to **generate** these scenarios and measure the whole models uncertainties



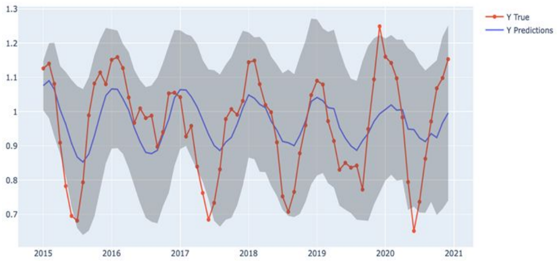
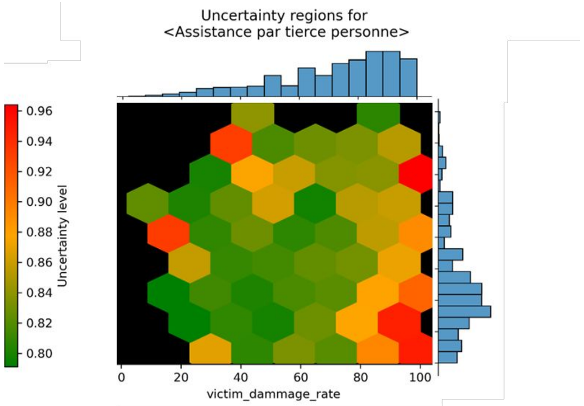


Figure 14: SWI prediction using Bayesian LSTM



[5] N. G. Polson, V. Sokolov et al., (2017) Deep learning: a Bayesian perspective, Bayesian Analysis, vol. 12, no. 4, pp. 1275–1304. https://arxiv.org/pdf/1706.00473.pdf
[6] Y Gal, (2016) Uncertainty in Deep Learning, http://www.cs.ox.ac.uk/people/yarin.gal/website//thesis/thesis.pdf

3.2 EXHAUSTIVITY - AUGMENTATION PROCESS

- Randomly **drop regions** from the original dataset and train both SDV and BNN
- We **inject** synthetic data in empty regions (black regions) using conditional generator (CTGAN)
- The trained BNN model will predict the **uncertainties** on injected synthetic data
- Compare the **measured** uncertainties between the synthetic and the real data

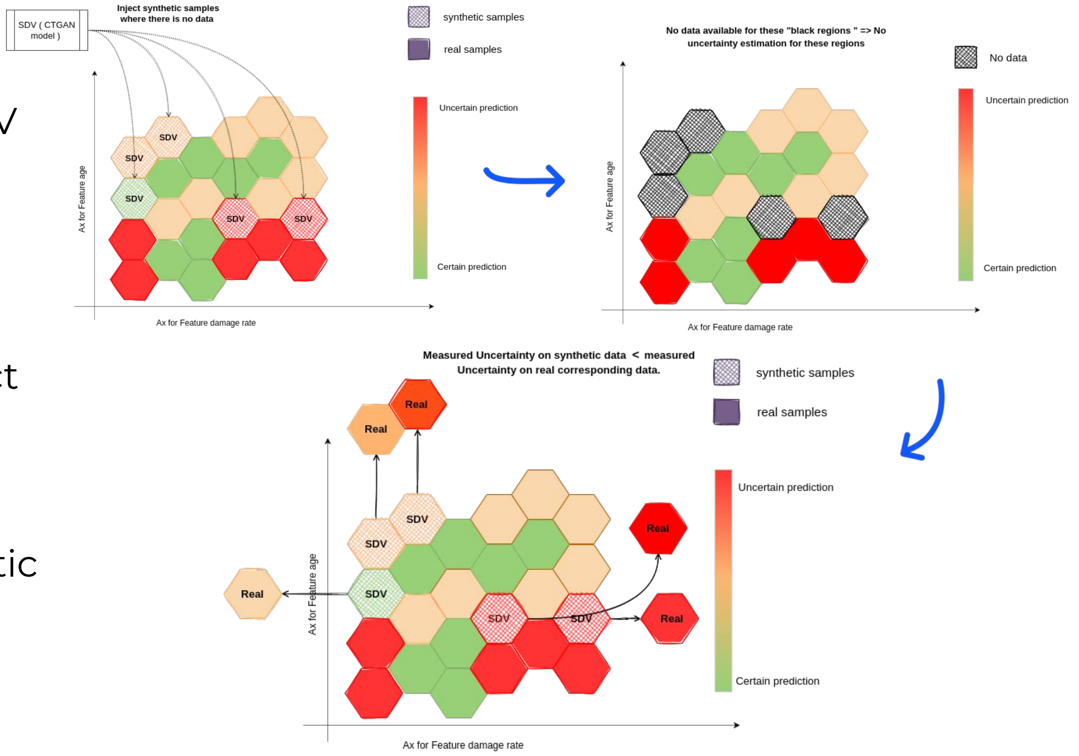
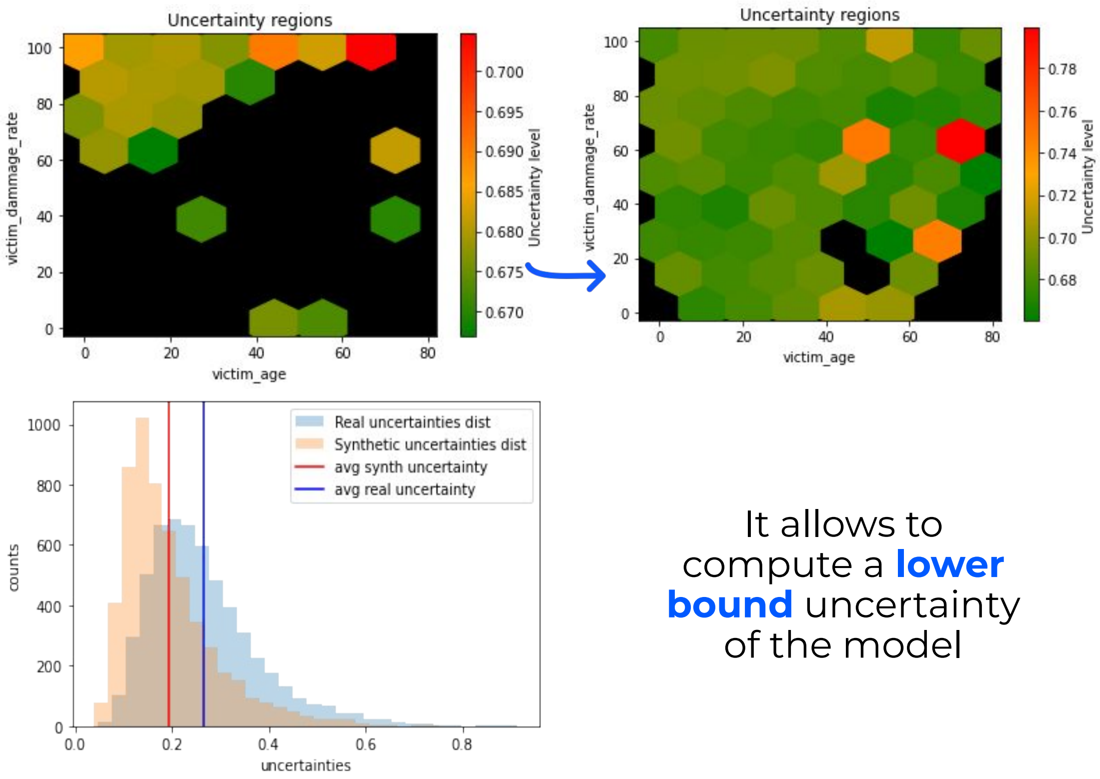


Figure 15: Synthetic data process of completion

3.2 EXHAUSTIVITY - IN PRACTICE

- Train of **BNN** and **CTGAN** models
- Use of CTGAN to inject synthetic data in the **black regions** and of the BNN to estimate the associated **uncertainties**
- We represent synthetic uncertainty distribution and **compare** it with real data uncertainty distribution
- In any cases, we observed that synthetic uncertainty was **shifted** to the left

Figure 16: Uncertainty generation using synthetic data



It allows to compute a **lower bound** uncertainty of the model

4. CONCLUSION AND PERSPECTIVES

4.1 CONCLUSION & PERSPECTIVES

- Synthetic Data implementation is helpful to handle data issues, specifically for insurance use cases:
  - **About quality**: a good imputer in addition to common techniques
  - **About exhaustivity**: a good approach to back test model scope

- Future perspectives:
  - Use other techniques for generation, such as TVAE
  - Tool testing (excel file sensitivity) Examples: S2, non life reserving
  - Other task types: NLP, (aerial) image
  - Wide field of investigation and many libraries: nlpaug, sdv, faker, mimesis, datasynthetizer, scikit learn, bifrost

Thank you!

APPENDIX - REFERENCES

- [1] PEIXIANG Z., Generative Adversarial Network (GAN) Overview, 2018, Github, [Link]
- [2] Minsuk Kahng, Nikhil Thorat, Polo Chau, Fernanda Viégas, and Martin Wattenberg. "GAN Lab: Understanding Complex Deep Generative Models using Interactive Visual Experimentation." Jan. 2019. https://arxiv.org/abs/1809.01587
- [3] Lei et al, Oct 2019. Modeling tabular data using conditional GAN. https://arxiv.org/abs/1907.00503
- [4] Daniel J. Stekhoven, Peter Bühlmann, MissForest—non-parametric missing value imputation for mixed-type data, https://doi.org/10.1093/bioinformatics/btr597
- [5] N. G. Polson, V. Sokolov et al., (2017) Deep learning: a Bayesian perspective, Bayesian Analysis, vol. 12, no. 4, pp. 1275–1304. https://arxiv.org/pdf/1706.00473.pdf
- [6] Y Gal, (2016) Uncertainty in Deep Learning, http://www.cs.ox.ac.uk/people/yarin.gal/website//thesis/thesis.pdf

APPENDIX - OTHER REFERENCES

- Xu L, Skoularidou M, Cuesta-Infante A, Veeramachaneni K. Modeling tabular data using conditional gan. Advances in Neural Information Processing Systems. 2019;32. https://arxiv.org/abs/1907.00503
- Zhao, Zilong, Aditya Kunar, Robert Birke, and Lydia Y. Chen. "CTAB-GAN+: Enhancing Tabular Data Synthesis." arXiv preprint arXiv:2204.00401 (2022). https://arxiv.org/abs/2204.00401
- Bourou, Stavroula, Andreas El Saer, Terpsichori-Helen Velivassaki, Artemis Voulkidis, and Theodore Zahariadis. "A review of tabular data synthesis using gans on an ids dataset." Information 12, no. 09 (2021): 375. https://www.mdpi.com/2078-2489/12/9/375/pdf
- Salim Jr, Ally. "Synthetic patient generation: A deep learning approach using variational autoencoders." arXiv preprint arXiv:1808.06444 (2018). https://arxiv.org/ftp/arxiv/papers/1808/1808.06444.pdf
- Benali, Fodil, Damien Bodénès, Nicolas Labroche, and Cyril de Runz. "Mtcopula: Synthetic complex data generation using copula." In 23rd International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP), pp. 51-60. 2021. https://hal.archives-ouvertes.fr/hal-03188317/document
- Brenninkmeijer, Bauke, A. de Vries, E. Marchiori, and Youri Hille. "On the generation and evaluation of tabular data using GANs." PhD diss., Radboud University, 2019. https://www.ru.nl/publish/pages/769526/z04_master_thesis_brenninkmeijer.pdf

APPENDIX - OTHER REFERENCES

- Houssou, Regis, Mihai-Cezar Augustin, Efstratios Rappos, Vivien Bonvin, and Stephan Robert-Nicoud. "Generation and Simulation of Synthetic Datasets with Copulas." arXiv preprint arXiv:2203.17250 (2022). https://arxiv.org/pdf/2203.17250.pdf
- Xu, Lei. "Synthesizing tabular data using conditional GAN." PhD diss., Massachusetts Institute of Technology, 2020. https://dai.lids.mit.edu/wp-content/uploads/2020/02/Lei_SMThesis_neo.pdf
- Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." Advances in neural information processing systems 27 (2014). https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html
- Little, Claire, Mark Elliot, Richard Allmendinger, and Sahel Shariati Samani. "Generative Adversarial Networks for Synthetic Data Generation: A Comparative Study." arXiv preprint arXiv:2112.01925 (2021). https://unece.org/sites/default/files/2021-12/SDC2021_Day2_Little_AD.pdf
- Xu, Lei, and Kalyan Veeramachaneni. "Synthesizing tabular data using generative adversarial networks." arXiv preprint arXiv:1811.11264 (2018). https://arxiv.org/abs/1811.11264
- Zhang, Jun, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. "Privbayes: Private data release via bayesian networks." ACM Transactions on Database Systems (TODS) 42, no. 4 (2017): 1-41. https://dl.acm.org/doi/pdf/10.1145/3134428

APPENDIX - METHODS - Gaussian Copulas

- Copulas allows to isolate the dependency structure of a set of variables in a multivariate distribution.
  - We can construct any multivariate distribution by separately specifying the marginal distributions and the copula.
  - Works with numerical or categorical features (after performing an encoding).
  - Find marginal distribution for each variable using MLE or empirical estimator so it preserves marginal distributions.

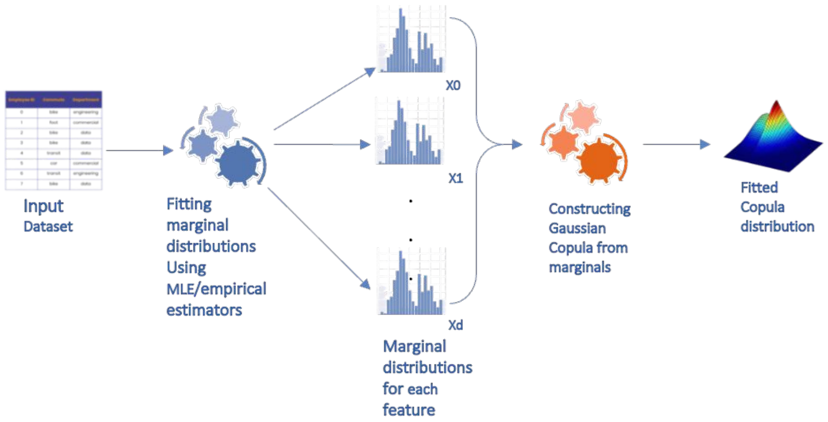


Figure: fitting a copula for a data table process

APPENDIX - METHODS - Bayesian Networks

- A Bayesian network is a graphical model of the joint probability distribution for a set of variables.
  - Two components: a graphical structure and a set of conditional probability distributions.
  - Search for a suitable network structure and probability distribution for a given dataset and then fit it to the data.
  - Generate differentially private synthetic data (make privacy concerns high priority)

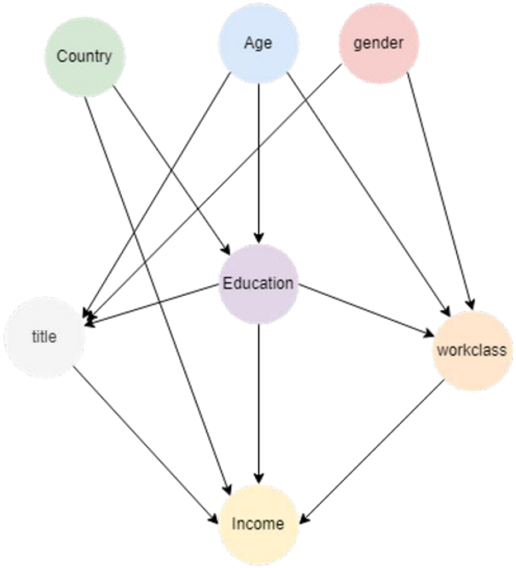


Figure: an example of what a Bayesian network looks like (authors).

APPENDIX - METHODS - Comparison of generation methods

- We used **SDGym*** library to evaluate the effectiveness of using synthetic data to train machine learning algorithms on different tasks. We use four datasets (Adult, Census and Covtype from UCI Machine learning and Credit from Kaggle) to generate corresponding synthetic datasets.

Dataset	size	Attributes	Continuous	Binary	Multi-class	task
Adult	32561	15	6	2	7	classification
Census	299385	41	7	3	31	classification
Covtype	581012	55	10	44	1	classification
Credit	284807	30	29	1	0	classification

Table: Used datasets characteristics (we used those provided in SDGym https://github.com/sdv-dev/SDGym/tree/master/results)

*** SDGym** is a benchmarking library developed by the team who created the **SDV** library .

APPENDIX - METHODS - Comparison of generation methods

- We trained then different classification models (Decision trees, AdaBoost and MLP) on real training data, and evaluating them on real test data. The Identity method corresponds to real training data.

Method	CovType			Credit		Adult		Census	
	Accuracy	F_1-score (micro/macro)		Accuracy	F_1-score	Accuracy	F_1-score	Accuracy	F_1-score
Identity	0,758886	0,652621	0,758886	0,992483	0,545017	0,824425	0,663005	0,905330	0,463875
Gaussian copula	0,506743	0,182262	0,506743	0,998250	-	0,779675	0,198041	0,934769	0,132829
PrivBayes	0,468946	0,214713	0,468946	0,960120	0,010973	0,795191	0,428731	0,903208	0,244719
CTGAN	0,581583	0,329751	0,581583	0,993329	**0,523338**	0,78525	0,606637	0,890426	**0,387663**
TableGAN	-	-	-	0,995366	0,27029	0,791183	0,352537	**0,936630**	0,272120
TVAE	**0,654793**	**0,456446**	**0,654793**	**0,99825**	-	**0,803008**	**0,618866**	0,934451	0,382320

Table: results of evaluation (we used those provided in SDGym https://github.com/sdv-dev/SDGym/tree/master/results)