Fill In Your Title

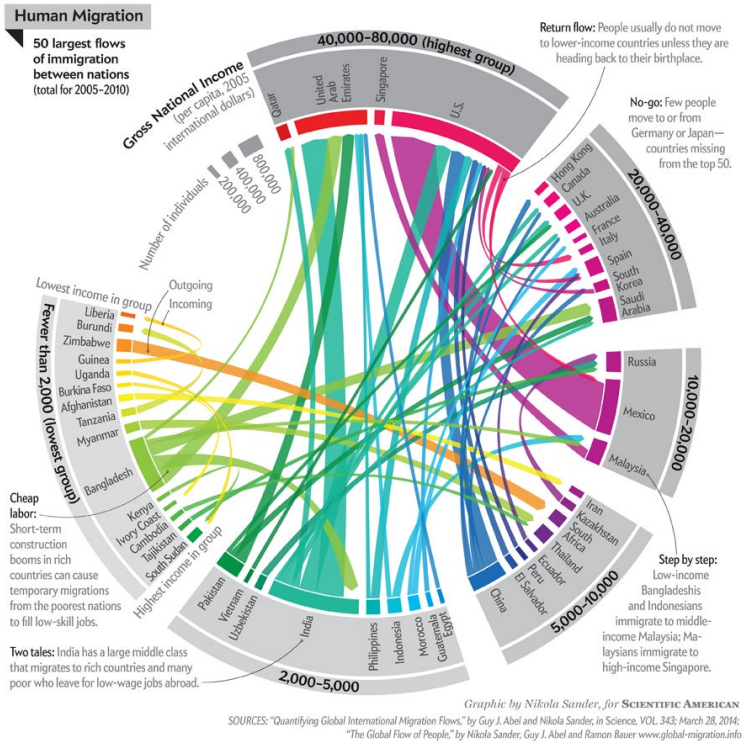
Fill In Your Sub-Title

submitted in partial fulfillment for the degree of master of science

Your Name
Your Student ID

Master Information Studies
data science
Faculty of Science
University of Amsterdam

Submitted on Fill In The Date In Format DD.MM.YYYY



	UvA Supervisor	External Supervisor
Title, Name	UvA Supervisor	External Supervisor
Affiliation	UvA Supervisor	External Supervisor
Email	supervisor@uva.nl	supervisor@company.nl

Amsterdam Data Science

The Awesome Company ™

ABSTRACT

Write your abstract here.

KEYWORDS

keywords, belong, here, with, commas, like, this

GITHUB REPOSITORY

https://github.com/you/your_awesome_thesis_repo

1 INTRODUCTION

The aim of this research is to enhance the accuracy and dependability of customer claim frequency prediction for motor third party liability (MTPL) insurance. Together with severity models, frequency models assess the amount of money a customer is liable to pay due to risk exposure from accidents in MTPL insurance. At present, generalized linear models (GLM) are utilized to predict customer exposure risk, but their effectiveness is limited as they require a significant amount of reliable customer exposure risk data, which can be scarce for certain groups, such as motorcyclists and truck drivers. This shortcoming is a major cause of inadequate pricing strategies, negative bias, and non-competitive pricing for insurers.

First steps towards solving this challenge have been already done by promising previous research [1, 4], who researched the use of different types of generative adversarial networks (GANs) to generate synthetic customer exposure risk data used to train improved GLMs. The most promising of all approaches included using a Multi-categorical Wasserstein GAN with a gradient penalty (MC-WGAN-GP). Our research hopes to further improve the quality of the MC-WGAN-GP by adding expert input into the training steps of both models (i.e., MC-WGAN-GP and GLM). Expert input will be provided in two forms into the model, namely before training the MC-WGAN-GP by generating new, expert-based variables, and after training by selecting generated data to be dropped from the generated dataset to reflect the real data distribution. The proposed approach aims to produce more reliable generated datasets that could be used to build better GLMs for claim frequency models in vehicle insurance.

The proposed research has the potential to contribute to the development of more accurate and reliable claim frequency models, which are less dependent on the availability of high amounts of internal data. With limited availability of real data, the proposed MC-WGAN-GP structure can generate synthetic data to enrich it. By generating synthetic data, insurers can create more accurate models to price their products and better understand the risks associated with a particular portfolio. This would enable insurance companies to enter new markets without having to obtain large amounts of data for computation of insurance pricing models, ultimately reducing the time and cost involved in creating new products.

Moreover, the research could also contribute to the development of more accurate and reliable methods for insurance pricing and reserving, which would benefit both insurers and policyholders. With more accurate risk assessment, insurers can price their products more accurately. Insurance companies can also use this research to gain a competitive advantage by pricing their products more accurately than their competitors. The proposed research also aims to test the general effectiveness of Neuro-symbolic input to GANs in the field of insurance, contributing to the field of data science and actuarial studies in general. In addition, the research could potentially lead to further development and refinement of the GAN models for other industries, creating new opportunities for data-driven decision making.

Overall, the potential use cases of this research are significant, including improving the accuracy and reliability of insurance customer exposure risk models, enabling insurance companies to enter new markets, and developing a more accurate and reliable method for generating synthetic data for insurance models. The primary objective of the research is to bridge the gap in the field of actuarial studies by exploring the potential of GANs for generating synthetic data to enhance the accuracy and reliability of vehicle insurance customer exposure risk models. The findings from this research could have a significant impact on the insurance industry and contribute to the development of more effective data-driven decision making in the field of actuarial studies.

With the increased interest in applying machine learning (ML) and predictive modeling techniques across all fields of actuarial science in recent years, access to data is becoming more important. Having access to realistic data from insurers allows researchers to tackle more practical problems and validate newly developed methodologies. If the data is also publicly available, it allows researchers and companies to open source their methodologies, which encourages others to build upon existing work. Furthermore, having a common collection of datasets for each research question allows the community to define the state of the art, benchmark new workflows, and measure progress. Of course, much of the data in the industry is confidential and proprietary. While there are publicly available datasets, new datasets are hard to come by. Even if data owners are willing to share anonymized datasets, the effort involved in obsfucating the data and navigating beauracracy may prevent them from doing so. We posit that, if there is an easy way for data owners to create “fake”, or synthesized, data that have characteristics similar to the real data, it would reduce friction in data disclosure. While synthetic data generation is an active area of research in the broader ML community, with many recent results (see, e.g., Choi et al. (2017), Park et al. (2018), and Xu et al. (2019)), research on synthesizing insurance datasets in particular is scarce. One notable example is Gabrielli and Wüthrich (2018), which describes a methodology for fitting neural networks to claims history data. The authors provide a fitted model for researchers to generate data, and the model has been implemented as an R package.1 However, it does not provide an easy way for a data owner to develop a new data generator from a different portfolio of claims. In this paper, we propose a workflow to train a neural network-based data synthesizer using confidential data and generate data from the trained synthesizer. We utilize the CTGAN architecture proposed by Xu et al. (2019), which is based on generative adversarial networks (GAN) (Goodfellow et al., 2014), and introduce modifications along with pre- and post-processing transformations specific to insurance datasets. We introduce an extension of the ML efficacy

evaluation methodology from the CTGAN paper utilizing cross-validation, and evaluate our workflow on two publicly available datasets using this methodology. To promote adoption, we provide an R package and code templates for researchers and data owners to use. The remainder of the paper is organized as follows. Section 2 provide brief overviews of GAN and CTGAN and introduces our workflow, Section 3 applies the workflow to two publicly available datasets, describes our evaluation methodology for ML efficacy, and evaluates the synthesized datasets with it, Section 4 discusses the data disclosure workflow and data privacy considerations, and Section 5 concludes.

2 RESEARCH QUESTION

This research aims to investigate whether GANs can be used to improve the accuracy of vehicle insurance customer exposure risk models by generating synthetic data and if provision of external actuarial expert input to the GANs can improve their quality. The research question can be stated as:

How can expert input be used to improve the quality of synthetic data generated by a generative adversarial network (GAN) such that it improves the accuracy of claim frequency models for MTPL?

To achieve this, the research will test three hypotheses:

(1) *Data Generation: Enriching real customer risk data with generated data by the MC-WGAN-GP improves the prediction quality of the final claim frequency GLM.*
(2) *Expert Input: Through addition of external actuarial expert input to the training of the GLM the quality of the GLM can further be improved, compared to pure data generation case.*
(3) *Data Generation and Expert Input: Through addition of external actuarial expert input to the training and data generation process of the MC-WGAN-GP and the GLM, the quality of the final GLM can further be improved, compared to the previous two cases.*

3 RELATED WORK

Write about your related work here. Make clear to which key papers you will compare your eventual results. This can be done from the perspective of methods used, the task at hand and the addressed domain.

3 RELATED WORK84 Given that our research connects different research areas with each85 other (i.e., neuro-symbolic AI, GANs in Underwriting), the related86 work section is categorized into these two different categories.87 1 3.1 Deep Learning in Underwriting88 There is a growing body of research on the development and im-89 plementation of models to better understand and predict customer90 exposure risk in vehicle insurance. Historical approaches towards91 modeling vehicle insurance exposure risk involved computation of92 exposure averages by customer groups which were later replaced by93 generalized linear models (GLM). These were gradually improved94 through the years, for example, by experimenting with different un-95 derlying data distributions assumed by the model [2]. More recent96 research started to be interested in using machine learning mod-97 els such as random forests for modeling risk exposure in vehicle98 insurance [9].99 More recently, actuarial research has caught interest in deep100 learning methods. For example, [8] and [11] analyze the potential101 of nesting a generalized linear model (GLM) in a neural network to102 better predict motor third-party liability claims. Also, [10] analyzes103 the potential of neural networks for chain-ladder reserving. Inter-104 estingly, only two applications of GANs in the actuarial sciences105 were researched to our knowledge.106 First, [5] suggested a workflow for synthesizing insurance datasets107 leveraging CTGAN. Second, [1] applied Generative Adversarial108 Networks (GANs) to synthesize property and casualty ratemaking109 datasets in the insurance industry. They proposed a method for110 generating synthetic datasets that can be used to train machine111 learning models without disclosing the original customer exposure112 risk data of an insurance's customers. Finally, [6] have used GAN113 models to impute missing data and correct incorrect inputs for114 automated underwriting models.115 Additionally, notable research on the replication of seldom events,116 a topic heavily covered in underwriting research, is the approach117 by [4], who developed FinGAN, a GAN for oversampling fraud -118 a seldom event - for further training of fraud detection models in119 banking and insurance.120 3.2 Neuro-Symbolic AI in Insurance121 The field of neuro-symbolic AI is fairly young, and therefore, only122 limited research has been conducted on the topic of Neuro-symbolic123 AI in insurance. However, certain noteworthy approaches have been124 conducted in the field of Neuro-Symbolic AI in Finance. For exam-125 ple, [3] have succeeded in improving the assessment of bank-loan126 applications using Neurules, neuro-symbolic rules that combine127 a symbolic representation (production rules) with a connectionist128 representation (adaline unit). Although this research is focused on129 banking, its insights can be easily transformed to the insurance sec-130 tor as the process of bank-loan applications is similar in structure131 to the insurance process of claim approval and can, therefore, be132 modeled similarly. Research focused on the use of Neuro-Symbolic133 approaches in insurance was conducted by [7], who presented an134 intelligent approach to the automated assessment of life insurance135 applications, which is based on an integration of neurule-based136 with case-based reasoning.137 3.3 Related Work Conclusion138 This section discussed related research on the use of machine learn-139 ing and neuro-symbolic AI in the field of insurance. The proposed140 thesis project builds on and contributes to this existing literature by141 combining the use of GANs and machine learning models for pre-142 dicting customer risk exposure with neuro-symbolic AI techniques.143 To the best of our knowledge, the use of GANs for generating144 synthetic data to enhance the performance of machine learning145 models for predicting risk exposure and the use of neuro-symbolic146 approaches towards data generation in insurance have not been147 extensively explored yet. Thus, the proposed project aims to fill148 this gap in the literature and advance the field of insurance under-149 writing.

4 METHODOLOGY

5 METHODOLOGY

5.1 Research Design

We conducted a study to examine the effectiveness of combining expert input, GANs, and GLMs for predicting claim frequency using five different modeling approaches. The five pipelines (see Figure

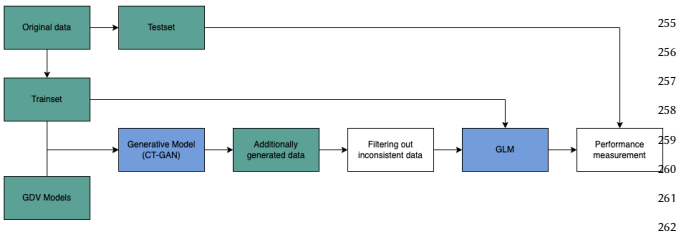


Figure 1: Structure of the pipeline (green: data, blue: models, white: human interaction)

1) were executed similarly, except for the use of expert input and data generation using GANs. The first pipeline served as a baseline and involved a regression model fitted on preprocessed data. In the second pipeline, a GAN was used for data generation and its generated data combined with real data to train the GLM. In the third pipeline, expert input was added before training the GAN for improved quality in the form of relationship estimation between frequency and the various independent variables. In the fourth pipeline, expert input was added before and after training the GAN for improved quality, including relationship estimation and distribution adjustment. The fifth pipeline included relationship estimation before training the GLM and no use of the GAN.

5.2 Expert Input

Expert input consisted of two parts. First relationship estimation before the training of the GAN and GLM and second adjustment of distribution after generating data with the GAN. Relationship estimation involved creating plots of claim frequency averages and rolling averages of claim frequency predictions for categorical and continuous variables, respectively. The distribution adjustment involved creating plots of the distribution for each variable for the original as well as the generated data. The actuary then identified where the data was wrongly distributed and removed random cases of the affected category/value to correct the distribution.

5.3 Data

We utilized the French motor third-party liability (MTPL) insurance portfolio data available on the OpenML platform [2]. The data consisted of multiple car insurance policies, each associated with 12 variables, including policy number (IDpol), claim count (ClaimNb), total yearly exposure (Exposure), area code (Area), power of the car (VehPower), age of the car in years (VehAge), age of the driver in years (DrivAge), bonus-malus level between 50 (bonus) and 230 (malus) (BonusMalus), car brand (VehBrand), fuel type (VehGas), density of inhabitants per km2 in the city of the living place of the driver (Density), regions in France (Region). Since the dataset is publicly available, we refer to [5] for a descriptive analysis of the data.

5.4 Data Preprocessing

For data preparation, we followed previous research [1, 5] and devised a common algorithm for all pipelines.

For the common data preparation, we included the following transformations on the data:

- Exposure: capped values exceeding one year
- ClaimNb: Converted to a categorical variable for GAN training, converted back after training
- DrivAge: Added new categorical variable DrivAgeCat which consists of 7 bins ($[18, 21)$, $[21, 26)$, $[26, 31)$, $[31, 41)$, $[41, 51)$, $[51, 71)$, $[71, \infty)$
- VehAge: Added new categorical variable VehAgeCat which consists of three bins ($[0, 1)$, $[1, 10]$, $(10, \infty)$)
- VehPower: Added new categorical variable VehPowerCat which consists of bins for increasing values of the variable; as decribed in [5]
- Density: Applied logarithmic transformation
- Area: Added new continuous variable AreaCont based of transformation of Area into continuous increasing variable
- VehPower: Transformed to a categorical variable with values equal to or greater than 9 were merged into a single categorical class
- BonusMalus: Capped at values exceeding 150
- IDpol: Dropped
- Exposure: Not used for GLM training

The resulting adjusted dataset consisted of X samples with seven numerical variables (ClaimNb, DrivAge, VehAge, VehPower, Density, AreaCont, BonusMalus) and categorical (ClaimNb, VehPower, VehBrand, VehGas, and Region) variables.

Data will be split into training and test set (75% and 25% respectively)[1].

5.5 General Linear Model (GLM)

In accordance with [5], the claim count per individual is assumed to follow a Poisson distribution. Thus, a Poisson generalized linear model (GLM) with a log-linear shape in the continuous feature components is utilized to model frequency data, with the canonical log-link function provided by scikit-learn. The regression function is calculated as $\lambda : \mathcal{X} \to \mathbb{R}_+$ by

$$\boldsymbol{x} \mapsto \log \lambda(\boldsymbol{x}) = \beta_0 + \sum_{l=1}^{d} \beta_l x_l,$$

where the parameter vector is $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_d)' \in \mathbb{R}^{d+1}$. Additionally, it is assumed that for $i \geq 1$,

$$N_i \overset{\text{ind.}}{\sim} \text{Poi}\left(\lambda\left(\boldsymbol{x}_i\right) v_i\right).$$

5.6 Multicategorical Wasserstein Generative Adversarial Model with Gradient Penalty (MC-WGAN-GP)

Let us first introduce the general framework of generative adversarial networks. The training of a GAN is a game between two competing networks: the generator and the discriminator. The generator G is a neural net with parameter vector θ_g that takes in argument a vector of random noise Z with distribution F_z, and maps it to the space of the data we wish to model. Usually, the components of the vector Z are independent standard Gaussian random variables, and the dimension of Z is lower than the that of the data. The resulting $G\left(Z; \theta_g\right)$ is a fake data point, and its distribution is denoted by F_g.

[1] wherever applicable, we use a seed of 1 for random operations

The goal of the training procedure is therefore to find a good approximation F_g of the unknown distribution of a true data point X, denoted F_x. To achieve this goal, a competing network, the discriminator D with parameter vector θ_d, learns to determine whether a data point is real or fake. To this end, the parameters θ_d of D are trained to maximize the expected score of a real data point $\mathrm{E}_X \{D(X;\theta_d)\}$ and to minimize the expected score of a synthetic data point $\mathrm{E}_Z \left[D\left\{G\left(Z;\theta_g\right);\theta_d\right\}\right]$. To achieve the goal of generating realistic data points, the parameters θ_g of the generator are trained to maximize the discriminator's score on a fake data point $\mathrm{E}_Z \left[D\left\{G\left(Z;\theta_g\right);\theta_d\right\}\right]$. Combining the two problems together, the two networks aim to solve

$$\min_{\theta_g}\max_{\theta_d} \mathrm{E}_X \left[\log\left\{D\left(X;\theta_d\right)\right\}\right] + \mathrm{E}_Z \left[\log\left[1 - D\left\{G\left(Z;\theta_g\right);\theta_d\right\}\right]\right]$$

This optimization problem amounts to minimizing the Jensen-Shannon divergence between F_x and F_g. In practice, this leads to serious convergence issues, partly solved by training D and G in turn with minibatches.

To solve some of the convergence issues, Arjovsky et al. (2017) advocate the use of the Wasserstein-1 distance between F_x and F_g, that is, they consider the problem

$$\min_{\theta_g}\max_{D\in\mathcal{D}} \mathrm{E}_X\{D(X)\} - \mathrm{E}_Z \left[D\left\{G\left(Z;\theta_g\right)\right\}\right]$$

where $\mathcal{D}$ is the set of 1 -Lipschitz functions. This change in the objective function leads to the Wasserstein generative adversarial network, or WGAN. The discriminator in a WGAN is called the critic, as it is real valued rather than a binary classifier. The WGAN is depicted schematically in Figure 1 for policyholder claim data X. The black arrows represent the forward flow of information in the network, while the colored arrows represent the flow of the training process for the generator (orange) and the critic (blue).

Some tactics are needed to enforce the Lipschitz constraints on D. In this regard, the gradient penalty (GP) developed by Gulrajani et al. (2017) greatly improves the training of the WGAN. In their WGAN-GP, the authors take advantage of the fact that a differentiable Lipschitz function has gradients with norm at most 1 everywhere. A tuning parameter $\lambda > 0$ is introduced, and the objective of the WGAN-GP is

$$\min_{\theta_g}\max_{\theta_d} \mathrm{E}_X \left\{D\left(X;\theta_d\right)\right\} - \mathrm{E}_Z \left[D\left\{G\left(Z;\theta_g\right);\theta_d\right\}\right] + \lambda \mathrm{E}_{\hat{X}} \left[\left\{\left\|\nabla_{\hat{x}} D\left(\hat{X};\theta_d\right)\right\|_2 - 1\right\}^2\right]$$

where $\hat{X} \overset{d}{=} UX + (1-U)G\left(Z;\theta_g\right)$, and U is uniformly distributed on the interval $(0,1)$, so that the distribution $F_{\hat{x}}$ of $\hat{X}$ is obtained by sampling uniformly along lines between pairs of points sampled from F_x and F_g. For details on the motivation, the reader is referred to Gulrajani et al. (2017).

In practice, if $m \in \mathbb{N}$ is the size of the minibatch with observations $x_1, \ldots, x_m$, random noise vectors $z_1, \ldots, z_m$ and independent uniform samples $u_1, \ldots, u_m$, then we let $\hat{x}_i = u_i x_i + (1-u_i)\, G\left(z_i;\theta_g\right)$ and the discriminator loss is approximated by

$$\mathcal{L}_d = \frac{1}{m}\sum_{i=1}^{m} -D\left(x_i,\theta_d\right) + D\left\{G\left(z_i;\theta_g\right);\theta_d\right\} + \lambda\left\{\left\|\nabla_{\hat{x}_i} D\left(\hat{x}_i;\theta_d\right)\right\|_2 - 1\right\}^2$$

while the generator loss is simply

$$\mathcal{L}_d = \frac{1}{m}\sum_{i=1}^{m} -D\left\{G\left(z_i;\theta_g\right);\theta_d\right\}$$

Note that higher values of the critic D indicate fake samples.

In order to accommodate the multi-categorical nature of our model, we make use of an addition to the WGAN-GP which allows it to make improved multi-categorical choices.

Camino et al. (2018) modified the generator of the WGANGP so that, after the model output, there is a dense layer in parallel for each categorical variable followed by a softmax activation function. Then, the results are concatenated to yield the final generator output. As in Camino et al. (2018), our generator's architecture has one dense layer with dimension matching the number of levels for each multi-categorical variable. We also add one dense layer with linear activation and dimension nc which is equal to the number of continuous variables. The architecture of the generator and the critic in our multi-categorical and continuous WGAN-GP, or MC-WGAN-GP, is depicted in Figure 2. Further details about hyperparameter optimization are available in Appendix A.

For each configuration of the Wasserstein GAN, the autoencoder (AE) and the GAN were trained independently one after the other, since the GAN requires the decoder. The training of the AE was done over 20,000 iterations, which was determined to be sufficient for convergence. Before feeding the preprocessed data to the network, the numerical features were normalized using min/-max normalization while the categorical variables were encoded using one-hot vectors. Unlike the more common way to encode N categories in N 1 dimensions, it was decided to use one dimension per category instead. Otherwise, the GAN would never generate the category not having a dimension of its own. The AE uses the binary cross entropy loss. The training of the GAN was done over 2M iterations. This value was determined empirically based on the results obtained. However, because of the known difficulty to train GANs (no stability guarantees), it could be adjusted depending on the configuration and the hyperparameters. The generator and discriminator use the zero-sum objective function as proposed by Arjovsky et al. (2017) for their learning. As recommended in this same paper, the discriminator was updated more often than the generator in order to train until optimality. Using a linear activation as the output layer of the discriminator and the RMSProp optimizer for the GAN are also recommendations from these authors that were applied. For both the AE and the GAN, the preprocessed dataset was split 2/3 for training and 1/3 for validation. The basic architecture of the networks (the number and sequence of layers) was not changed from Tantipongpipat et al. (2019). Unless stated otherwise, the activation functions used were always LeakyReLU with a negative slope of 0.2. Because they depend on the input size and some hyperparameters (such as the latent dimensions), the sizes of the layers vary from configuration to configuration. The design choices of using the ADAM optimizer with gradient penalty for the AE, choosing the absolute bounds to clip the values of the WGAN gradients and choosing layer normalization over batch normalization for the generator follow the recommendations of Gulrajani et al. (2017). The algorithm of the differentially private stochastic gradient descent (DP-SGD) can be found in Tantipongpipat et al.

(2019). For the differential privacy aspect, computing the L2 clipping norms of the gradients for the decoder and the discriminator was done as recommended by Abadi et al. (2016). Finally, before training the models used to obtain results, the hyperparameters were tuned using a random search.

As for the training, the tuning of the hyperparameters of the AE and the GAN was done in two stages. In a random search, each combination of hyperparameters tested is selected randomly from the chosen grid. The number of search iterations is set based on a time/resources compromise. The strategy was to run multiple searches instead of a single big one. Each time, the search space was narrowed for more fine-tuning. In all cases, the tuning was done in a non private way (=) in order to reduce computation time. For the AE, the hyperparameters tested were the minibatch size, compression dimension, learning rate, 1 and 2 parameters of the ADAM optimizer, and the L2 penalty of the weight decay for the optimizer. In first experiments, these last three hyperparameters did not affect the training significantly so it was decided to leave them at their usual default values (0.9, 0.999 and 0, respectively). To evaluate the performance of each combination, the validation and training losses were saved and sorted. The combination with the lowest final validation loss was chosen. A regular training was then done to confirm that it was not overfitting. For the GAN, the hyperparameters tested were the minibatch size, the latent dimension of the generator, the learning rate, the number of iterations of the discriminator before updating the generator once, the L2 penalty of the weight decay of the optimizer and the smoothing constant of the RMSProp optimizer. Once again, these last two hyperparameters of the optimizer did not affect the results significantly so they were left at 0 and 0.99 respectively. The evaluation of the performance of the GAN was not as straightforward as for the AE. Both the losses of the discriminator and the generator were plotted. Over time, it was found that the desired loss curve of the discriminator was one which dropped rapidly near 0 and that then converged to that value over training iterations. For the generator, a loss oscillating rather slowly around 0 (going positive for many thousands of iterations then going negative and so on) appeared a good indicator of performance. Fortunately, these tendencies could be spotted for training of only a few tens of thousands of iterations. Hence, to reduce computation time, the number of iterations for each combination was limited to 100,000.

In the MC-WGAN-GP model, we tuned the following hyperparameters.

- Loss Penalty - loss_penalty
- Generator Batch Norm Decay - gen_bn_decay
- Discriminator Batch Norm Decay - disc_bn_decay
- Generator L2 Regularization - gen_L2_reg
- Discriminator L2 Regularization - disc_L2_reg
- Learning Rate - learning_rate

We used a random search to explore the settings and decided on the values in Table A.2. Table A.2: Hyperparameter settings for the MC-WGAN-GP.

Hyperparameter	Explored Values	Chosen Value
$\text{loss}_penalty$	1, 5, 10, 20, 50	10
$\text{gen}_b n_d ecay$	0, 0.10, 0.25, 0.45, 0.50, 0.90	0.90
$\text{gen}_L 2_r eg$	0, 0.00001, 0.0001, 0.001, 0.01	0
$\text{disc}_L 2_r eg$	0, 0.00001, 0.0001, 0.001, 0.01	0
$\text{learning}_r ate$	0.001, 0.005, 0.01	0.01

Once a couple of potentially good combinations were identified in this manner, a full training over 2M iterations was done for each one. The generated samples of these trained models were then evaluated in respect to the univariate distributions of each variable (vs the real distributions). The predictions on the target ClaimNb of a random forest regressor and of a random forest classifier were also compared between the generated and the real samples. The combination giving the best overall results was kept. Because of the lack of stability of the GANs (even for the same hyperparameters and configuration), this best combination of hyperparameters was trained at least two more times with different seeds for the random number generator. The model of whichever run gave the best results was saved and used for the final results. For both the autoencoder and the GAN, for a given configuration, the hyperparameters that had the most impact on the results were the minibatch size and the learning rate. When training differentially private models, the values of the hyperparameters were the same as those of its corresponding non private configuration. To guarantee the privacy, both the autoencoder and the GAN were trained from scratch (i.e. their non private counterpart were not used at any point).

5.7 Evaluation

The study will evaluate the effectiveness of the proposed approach by comparing the performance of generalized linear models (GLMs) built on the original data and on synthetic data generated by the GAN. The GLMs will be evaluated using the Mean Absolute Percentage Error (MAPE) and Gini coefficient, which measure accuracy and variability of policyholder risks, respectively. Statistical tests, such as t-tests, will be performed to determine if the differences in the metric values are significant. The evaluation will be conducted in Python programming language and at a 95% confidence level.

Hypothesis 1 will be supported if the GLM built on enriched and real data performs significantly better on the holdout test set than the risk model built on only real data in at least one of the metrics and does not perform significantly worse on the others. Hypothesis 2 will be supported if the GLM built on enriched data with external coefficient input significantly performs better on the holdout test set than the risk model built on only real data in at least one of the metrics and does not perform significantly worse on the others.

To test hypothesis 3, the study will compare the average prediction quality metric delta between models built on GAN-enriched small data vs. non-enriched small data, and the average delta between models built on GAN-enriched big data vs. non-enriched big data. If there is a significant difference between the two differences, then hypothesis 3 is supported.

Overall, the evaluation will result in 6 different scenarios, each with 2 metrics (see table 1)

Data Availability	Enriching Method	Metrics
Whole trainingset	Not enriched	MAPE, GINI
Whole trainingset	CT-GAN	MAPE, GINI
Whole trainingset	CT-GAN + external coefficient input	MAPE, GINI
25% of trainingset	Not enriched	MAPE, GINI
25% of trainingset	CT-GAN	MAPE, GINI
25% of trainingset	CT-GAN + external coefficient input	MAPE, GINI

Table 1: Data used, GAN models compared, and metrics used for comparison

We evaluate the models' out-of-sample performance according to four metrics as in [3]: Average Poisson deviance loss (Dev), $\widetilde{\mathrm{R}}^2$, Lift, and Gini index. Dev is the average Poisson deviance loss as defined in Equation (3). We focus on this metric because loss minimization is our training target. Let y_i and $\hat{y}_i$ be the observed response and predicted value, respectively; then, Dev is defined as

$$D(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n} 2\left(y_i \log\frac{y_i}{\hat{y}_i} - y_i + \hat{y}_i\right)$$

where the i-th term of the right-hand side is equal to $2\hat{y}_i$ if $y_i = 0$.$\widetilde{\mathrm{R}}^2$ is a variant of Dev and measures the proportion of deviance reduction with regard to a null model.

$$\widetilde{\mathrm{R}^2} = 1 - \frac{D(y,\hat{y})}{D(y,e\bar{y})}$$

Here, $D(y,e\bar{y})$ represents the null deviance and $\bar{y} = \frac{\sum y_i}{\sum e_i}$ is the exposure weighted average of the observed frequency. Lift measures the ability of a model to distinguish between observations. We calculate it according to the following steps: (1) sort the policies in the order of predicted frequency from smallest to largest; (2) bin the policies into 10 groups with the same amount of total exposure; (3) in

the j-th bin($j = 1, 2, \ldots, 10$), calculate the average response and the average prediction as $\bar{y}_j$ and $\hat{y}_j$, respectively; (4) calculate the Lift of the model as $\bar{y}_{10}/\bar{y}_1$. Lift shows the local properties of the model, where a higher Lift illustrates that the model can better differentiate extreme values. The lift plot; that is, $\left(\hat{y}_j, \bar{y}_j\right)$ ($j = 1, 2, \ldots, 10$), reflects the overall bias between the model's result and the perfect prediction.

The Gini index is another metric to measure the discriminatory power of models. It produces more robust results than Lift, as it uses the full data rather than extreme points. There exist many variants of the Gini index and one we use is the normalized Gini index, denoted by Gini $^{\mathrm{a}}$ (see Zhou et al. [56]):

$$\text{Gini}^{\,\mathrm{a}} = \frac{\frac{\sum_{i=1}^{n} y_i r(\hat{y}_i)}{\sum_{i=1}^{n} y_i} - \sum_{i=1}^{n}\frac{n-i+1}{n}}{\frac{\sum_{i=1}^{n} y_i r(y_i)}{\sum_{i=1}^{n} y_i} - \sum_{i=1}^{n}\frac{n-i+1}{n}},$$

where $r(y_i)$ is the rank of y_i in the sequence $\{y_1, y_2, \ldots, y_n\}$ sorted in increasing order, $\sum_{i=1}^{n} y_i$ represents the sum of actual responses and $\sum_{i=1}^{n} y_i r(y_i)$ represents the sum of cumulative actual responses. Thus, the numerator of Equation (12) is the Gini index obtained by sorting the responses according to the predictions. The maximum possible Gini index, that is, the denominator of Equation (12), can be obtained when the actual responses are sorted by themselves. The ratio is what we call the normalized Gini index. Note that a larger Gini ^{a} index indicates a model with stronger discriminatory power to produce large/small predictions to observations with large/small actual responses.

The ordered Lorenz curve and Gini ^{b} index [74] are useful tools to compare models by analyzing the distributions of actual responses versus predictions. Let $B(x)$ and $P(x)$ be the base model prediction and the competing model prediction, respectively. Then, relativity $R(x)$ is defined as $R(x) = \frac{P(x)}{B(x)}$. We sort the policies in the order of the relativities $R(x_i)$ from smallest to largest. Next, compute two empirical distributions based on the same sort order. The ordered distribution of base prediction $B(x)$ is:

$$\widehat{D}_p(s) = \frac{\sum_{i=1}^{n} B(x_i)\,\mathrm{I}(R(x_i) \le s)}{\sum_{i=1}^{n} B(x_i)}$$

The ordered distribution of response y is:

$$\widehat{D}_r(s) = \frac{\sum_{i=1}^{n} y_i \mathrm{I}(R(x_i) \le s)}{\sum_{i=1}^{n} y_i}$$

The ordered Lorentz curve is the graph of $\left(\widehat{D}_p(s), \widehat{D}_r(s)\right)$. It visualizes the mismatch between responses and base predictions given knowledge of the competing predictions. The $y = x$ straight line, known as "the line of equality," indicates that the percentage of responses equals the percentage of base predictions. The Ginib index, computed as twice the area between the ordered Lorenz curve and the line of equality, quantifies the vulnerability of the base model to its competitor. A large Ginib index means that the competing model is useful for detecting the difference between the actual response and the base model prediction.

5.8 Implementation

The proposed approach will likely be implemented using Python programming language and the PyTorch deep learning library. The GAN will be trained using a CPU. The GLMs will be built built using the statsmodels Python package.

5.9 Ethical Considerations

Since the dataset is publicly available, we do not have to take any considerations into account regrading ethical data issues or data protection issues.

6 RESULTS

Write about your results here. Good captions to tables and/or figures are key.

7 DISCUSSION

Write your discussion here. Do not forget to use sub-sections. Normally, the discussion starts with comparing your results to other studies as precisely as possible. The limitations should be reflected upon in terms such as reproducibility, scalability, generalizability, reliability and validity. It is also important to mention ethical concerns.

8 CONCLUSION

Write your conclusion here. Be sure that the relation between the research gap and your contribution is clear. Be honest about how limitations in the study qualify the answer on the research question.

REFERENCES

[1] Marie-Pier Cote, Brian Hartman, Olivier Mercier, Joshua Meyers, Jared Cummings, and Elijah Harmon. 2020. Synthesizing property & casualty ratemaking datasets using generative adversarial networks. *arXiv preprint arXiv:2008.06110* (2020).

[2] Matthias Feurer, Jan N. van Rijn, Arlind Kadra, Pieter Gijsbers, Neeratyoy Mallik, Sahithya Ravi, Andreas Mueller, Joaquin Vanschoren, and Frank Hutter. [n. d.]. OpenML-Python: an extensible Python API for OpenML. *arXiv* 1911.02490 ([n. d.]). https://arxiv.org/pdf/1911.02490.pdf

[3] Yaqian Gao, Yifan Huang, and Shengwang Meng. 2023. Evaluation and interpretation of driving risks: Automobile claim frequency modeling with telematics data. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 16, 2 (2023), 97–119.

[4] Kevin Kuo. 2019. Generative synthesis of insurance datasets. *arXiv preprint arXiv:1912.02423* (2019).

[5] Alexander Noll, Robert Salzmann, and Mario V Wuthrich. 2020. Case study: French motor third-party liability claims. *Available at SSRN 3164764* (2020).

Appendix A FIRST APPENDIX

Put your appendices here.